



EASY BASIC PROGRAMS for the APPLE

Brian Flynn

Everything from games and money managers to statistical calculators, business forecasters, and household planners—these programs entertain, organize, and analyze. Over 40 programs for the home and office, ready to type in and run on your Apple II, II+, IIe, or IIfx computer.

A **COMPUTE! Books** Publication

\$14.95



EASY BASIC PROGRAMS for the APPLE

Brian Flynn

COMPUTE! Publications, Inc. 
One of the ABC Publishing Companies
Greensboro, North Carolina

The following programs were originally published in *COMPUTE!* magazine, copyright 1982, Small Systems, Inc.:

"Word Hunt" (March); "Math Man" (October—originally titled "MathMan")

The following programs were originally published in *COMPUTE!* magazine, copyright 1983, *COMPUTE!* Publications, Inc.:

"Weather Forecaster" (August); "Mosaic Puzzle" (October); "Calorie Cop" (December); "Paycheck Analysis" (December); "Utility-Bill Audit" (December)

Copyright 1985, *COMPUTE!* Publications, Inc. All rights reserved

Reproduction or translation of any part of this work beyond that permitted by Sections 107 and 108 of the United States Copyright Act without the permission of the copyright owner is unlawful.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-942386-88-4

COMPUTE! Publications, Inc., Post Office Box 5406, Greensboro, NC 27403, (919) 275-9809, is one of the ABC Publishing Companies and is not associated with any manufacturer of personal computers. Apple II, II+, IIe and IIC are trademarks of Apple Computer, Inc.

Contents

Foreword	v
Before You Begin	vii
1: Brainteasers	1
Guess a Word	5
Chomp	13
Math Man	
<i>Original Game Concept by Andy Hayes</i>	22
Matches	31
Word Hunt	
<i>Original Game Concept by Robert W. Baker</i>	39
Solitaire Checkers	51
Mosaic Puzzle	
<i>Original Game Concept by Bruce Jordan</i>	61
Spelling Quiz	71
Hi-Q	80
Beans	90
2: Personal Planners	99
IRA Planner	103
Car Payments	109
Mortgage Payments	113
College Planner	120
Effective Yield on Investments	129
Municipal Bond Buyer	133
Household Budget	142
Car Trek	151
Weather Forecaster	
<i>Original Program Concept by George W. Miller</i>	159
Calorie Cop	
<i>Original Program Concept by Gerald P. Graham</i>	170
Utility-Bill Audit	
<i>Original Program Concept by Larry L. Bihlmeyer</i>	178
Paycheck Analysis	
<i>Original Program Concept by Larry L. Bihlmeyer</i>	187
3: Number Crunchers	193
Sort	197
Random Number Generator	201
Numerical Integration	206
Derivative	214

Matrix Addition and Subtraction	219
Matrix Multiplication	227
Matrix Inversion	235
MatMan	242
4: Simple Statistics	253
Mean, Variance, and Standard Deviation	257
Relative and Cumulative Frequencies	262
Pie Plot	268
Frequency Plot	276
Analysis of Variance	284
Simple Correlation Coefficients	290
Simple Linear Regression Analysis	299
Multiple Linear Regression Analysis	308
General-Form Curve Fitter	322
5: Basics for Business	333
Database Management	337
Net Present Value	349
Internal Rate of Return	354
Least-Squares Forecasting	362
Time-Series Forecasting	372
Computer Cash Register	382
Appendix: Beginner's Guide to Typing In Programs . .	387
Index	389

Foreword

There's something for everyone in this collection of BASIC programs for the Apple series of computers. If you want to manage your personal finances better, you'll find budget organizers, IRA and college cost planners, investment and municipal bond analyzers, and other household applications. If you enjoy working with numbers and statistics, you can manipulate matrices or compute means, variances, and standard deviations. Calculus routines let you accurately compute the area under a curve. You can even create pie charts and bar graphs to display percentages.

Do you run a small business? Your computer can be an electronic cash register that automatically tallies sales tax. Better yet, it can forecast future trends in production, sales, income, or anything else. Do you want applications for the office? Then use the database manager or calculate the future worth of today's dollars.

Computers aren't just for work, though, and that's why this book includes ten challenging games. They'll keep you entertained while they make you think.

The instructions and explanations for each program are clear and concise. Many can easily be modified or customized for your own use. Best of all, the program listings are printed so that you can study them. After all, it's only a small leap from making minor changes in a program to creating your own programs.

Your Apple computer can be a powerful tool, both in the home and the workplace. And with these programs, it can become even more versatile.



Before You Begin

Each of the more than 40 programs in *Easy BASIC Programs for the Apple* has been tested and is ready to type in and run on your Apple II, II+, IIe, or IIfx. Each program is under 7500 bytes long, so don't worry if your machine has limited memory. The IIfx, for example, comes equipped from the factory with 128K (over 131,000 bytes) of memory.

The programs in this book are DOS 3.3 and ProDOS compatible. However, if you have ProDOS, you may prefer to use that.

Each program uses the 40-column display mode, so you don't need 80-column capability. If you are using a non-Apple monitor with a 40/80 character switch, just set it to 40.

The programs do take advantage of the upper- and lowercase printing available on the later Apples, like the IIfx. If your machine doesn't have lowercase characters available, key in the normal characters in place of the lowercase ones when entering programs.

Some of the programs in this book have a color mode available. If you are using a television rather than a color computer monitor, the colors may blur. If this happens, simply select the black-and-white option the next time you run the program.

To Save Time

Roughly two-thirds of these programs display numbers on the screen. To make the screen look neat, these programs often contain a BASIC subroutine that will round numbers and line them up on the decimal point. Lines 9000–9110 will make up the routine, and "IRA Planner" in Chapter 2 uses this routine.

Rather than enter the same lines over and over, just do it once. First, enter the subroutine and save it on a disk using a name like FORMAT. Then, whenever you encounter a program that contains this subroutine, load FORMAT before you begin entering the program. Lines 9000–9110 will then be in memory (to verify this, just LIST), saving you the time and effort of reentering them.



1 Brainteasers



Brainteasers

These short, yet challenging games are simple to program in BASIC because they have little or no graphic enhancements. They're not arcade-style games, and they're not meant to be. Instead of requiring quick reflexes and hyperactive hand/eye coordination, these brainteasers test your knowledge, concentration, and persistence.

Lady Luck, mistress of so many other contests, is banished here. In these games, what you accomplish is left up to you, rather than to some impersonal random number generator. But that's what makes each of these games so challenging, and so much fun to play again and again.

Because they don't depend on graphics to keep the player interested, these brainteasers are short and easy to type into your computer. And since they're in BASIC, you won't have any problem seeing how the program is structured when you look through the listings.

Before you begin entering these games, make sure you read the Appendix, "Beginner's Guide to Typing In Programs." All the programs are printed *exactly* as you should type them in.

The ten brainteasers in this first chapter are:

- **Guess a Word.** Try to guess a word that either your opponent or the computer selects.
- **Chomp.** Force one of your opponents to chomp on the only pear in an orchard of apples.
- **Math Man.** Test your ability to quickly add, subtract, multiply, and divide. A challenging game for all ages, with three levels of difficulty.
- **Matches.** You'll see between 7 and 17 matches in a row. Only the first is lit, however, and you need to pick it up before your opponent does.
- **Word Hunt.** Locate ten words as fast as you can in a maze of letters.
- **Solitaire Checkers.** Eliminate as many checkers as possible from a 7×7 board, containing 40 pieces.

- **Mosaic Puzzle.** A version of the old sliding-squares game. Rearrange the numbers of a 3×3 square, using as few moves and as little time as possible.
- **Spelling Quiz.** Everyone, from grade schoolers to grownups, can stand a bit of spelling practice.
- **Hi-Q.** The famous European solitaire game of finesse and foresight. Try to remove as many pegs as possible from a cross-shaped board, with only horizontal and vertical jumps allowed.
- **Beans.** An appetizing two-player game of logic. Nourish your intellect.

Some of these games are quite easy, while others are much more difficult to play well. Don't despair—it's one of the charms of brain teasers.

Guess a Word

Uncovering hidden words is the challenge in this one- or two-player game. Match your vocabulary against the computer's or your opponent's.

"Guess a Word" is a challenging vocabulary game. If you're playing alone, the computer randomly selects a word from the program's built-in dictionary. In a two-player game, each player selects words for the other. The object? Uncover the word by guessing its letters.

Random Seeding

Once you've typed in and saved Guess a Word, run it. The first screen display asks you for a number from 1 to 1000. Enter any number in that range. It's important that you use different numbers each time you load the game program. If you don't, the words will tend to appear in the same order time after time. It doesn't matter *which* number you use—just that it differs from session to session.

Mind or Machine

Next, select the one- or two-player option by typing Y if you want the computer to select the word, or N to enter a word yourself for an opponent to guess. The computer uses its own vocabulary list to choose words, while you enter your opponent's word when you play against a friend. The maximum length of a word when you're playing another person is 15 letters. It's probably best to agree beforehand to a limit on word length so that the contest is more equal.

Vocabulary Counts

You play by entering a letter you think might be in the word. If you're right, the computer reveals all locations of that letter, such as -a- in *cat* or -oo- in *goose*. If you're wrong, however, the computer removes one of the numbers at the top of the screen and adds the incorrect character to the screen display. You can choose up to seven incorrect letters before you've lost the round.

Here's a hint: Try entering vowels on your first couple of turns, particularly *i* or *e*. Needless to say, it's difficult to find a

word in the English language without vowels.

After each letter selection, you're asked if you want to attempt a guess. There's no penalty if you're wrong. If you enter the word correctly from the keyboard without making a guess, you still win. At that point you'll see the word and the number of incorrect guesses you used. On the other hand, if you've lost, you get to see the correct word. No matter what the result, playing another round is as simple as pressing the Y key. If you're ready to quit, enter N instead.

When you've memorized the computer's vocabulary, you may want to replenish it with selections of your own. All you have to do is enter new words in the DATA statements in lines 2450-2710. Remember to use words of 15 letters or less.

Guess a Word

```

100 REM GUESS A WORD
110 REM INITIALIZE
120 GOSUB 210
130 REM PLAY GAME
140 GOSUB 550
150 REM PLAY AGAIN
160 VTAB 23: HTAB 11: PRINT "Play again (Y
    /N) ? "; CHR$ (7);
170 GET S$
180 IF S$ = "Y" OR S$ = "y" THEN 140
190 IF S$ < > "N" AND S$ < > "n" THEN 17
    0
200 END
210 REM INITIALIZE
220 REM TITLE
230 GOSUB 270
240 REM INSTRUCTIONS
250 GOSUB 320
260 RETURN
270 REM TITLE
280 PRINT CHR$ (21): TEXT : HOME
290 VTAB 12: HTAB 14: PRINT "Guess a Word
300 FOR PAUSE = 1 TO 2000: NEXT
310 RETURN
320 REM INSTRUCTIONS
330 HOME
340 PRINT "The goal of this game is to 'Gu
    ess a
350 PRINT "Word.'"
360 PRINT
370 PRINT "If you're playing alone, I'll s
    elect
    
```


Brain teasers

```
380 PRINT "the word for you.
390 PRINT
400 PRINT "If two of you are playing, try
    selecting";
410 PRINT "words for each other.
420 PRINT
430 PRINT "Up to 7 wrong letters are allow
    ed before";
440 PRINT "you lose.
450 PRINT
460 PRINT "Please enter a number so that I
    can
470 PRINT "choose words randomly.
480 VTAB 16: HTAB 22: PRINT SPC( 20); CHR$
    (7)
490 VTAB 16: HTAB 1: INVERSE : PRINT "SEED
    (1 TO 1000)";: NORMAL
500 INPUT " = ? ";S$
510 V = VAL (S$)
520 IF V < 1 OR V > 1000 THEN 480
530 Z = RND ( - V)
540 RETURN
550 REM PLAY
560 REM SELECT WORD
570 HOME
580 PRINT "Would you like me to select the
590 VTAB 2: HTAB 1: PRINT "word (Y/N) ? ";
    CHR$ (7);
600 GET S$
610 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 590
620 IF S$ = "Y" OR S$ = "y" THEN GOSUB 67
    0
630 IF S$ = "N" OR S$ = "n" THEN GOSUB 75
    0
640 REM GAME
650 GOSUB 960
660 RETURN
670 REM COMPUTER SELECTS
680 RESTORE
690 C = INT ( RND (1) * 104) + 1
700 FOR I = 1 TO C
710 READ WD$
720 NEXT
730 L = LEN (WD$)
740 RETURN
750 REM PLAYER SELECTS
760 HOME
```

```

770 PRINT "Please enter the word that you
    want
780 PRINT "your opponent to guess. Up to
    15
790 PRINT "characters are allowed.
800 VTAB 5: HTAB 10: PRINT SPC( 40); CHR$
    (7)
810 VTAB 5: HTAB 1: INPUT "Word = ? ";W$
820 IF LEN (W$) < 3 THEN VTAB 23: HTAB 1
    1: PRINT "That's too easy !": GOTO 800

830 IF LEN (W$) > 15 THEN VTAB 23: HTAB
    11: PRINT "That's too long !": GOTO 80
    0
840 REM CONVERT TO CAPS
850 GOSUB 870
860 RETURN
870 REM CONVERT TO CAPS
880 WD$ = ""
890 L = LEN (W$)
900 FOR I = 1 TO L
910 L$ = LEFT$ ( RIGHT$ (W$,L - I + 1),1)
920 IF ASC (L$) > 90 THEN L$ = CHR$ ( ASC
    (L$) - 32)
930 WD$ = WD$ + L$
940 NEXT
950 RETURN
960 REM GAME
970 REM INITIAL POSITION
980 GOSUB 1110
990 REM DISPLAY
1000 GOSUB 1200
1010 REM ENTER LETTER
1020 GOSUB 1540
1030 IF H$ = WD$ THEN GUESS$ = WD$
1040 REM ENTER GUESS
1050 IF GUESS$ < > WD$ THEN GOSUB 1990
1060 IF N < 7 AND GUESS$ < > WD$ THEN 102
    0
1070 REM WINNER OR LOSER
1080 IF GUESS$ = WD$ THEN GOSUB 2280
1090 IF GUESS$ < > WD$ THEN GOSUB 2360
1100 RETURN
1110 REM INITIAL POSITION
1120 N = 0:GUESS$ = ""
1130 LTR$ = ""
1140 REM HIDDEN WORD
1150 H$ = ""
1160 FOR I = 1 TO L
1170 H$ = H$ + CHR$ (45)

```

```

1180 NEXT
1190 RETURN
1200 REM DISPLAY
1210 HOME
1220 REM BOX
1230 GOSUB 1270
1240 REM LABELS
1250 GOSUB 1410
1260 RETURN
1270 REM BOX
1280 INVERSE
1290 FOR I = 1 TO 39
1300 VTAB 1: HTAB I: PRINT CHR$ (32)
1310 VTAB 2: HTAB I: PRINT CHR$ (32)
1320 VTAB 3: HTAB I: PRINT CHR$ (32)
1330 VTAB 24: HTAB I: PRINT CHR$ (32);
1340 NEXT
1350 FOR ROW = 1 TO 23
1360 VTAB ROW: HTAB 1: PRINT CHR$ (32)
1370 VTAB ROW: HTAB 39: PRINT CHR$ (32)
1380 NEXT
1390 NORMAL
1400 RETURN
1410 REM LABELS
1420 VTAB 2: HTAB 14: PRINT " GUESS A WORD
"
1430 VTAB 8: HTAB 12: PRINT "Guesses Remai
ning
1440 COL = 14: INVERSE
1450 FOR J = 7 TO 1 STEP - 1
1460 VTAB 6: HTAB COL: PRINT J
1470 COL = COL + 2
1480 NEXT
1490 NORMAL
1500 VTAB 14: HTAB 3: PRINT "Bad letters
1510 VTAB 15: HTAB 3: PRINT "Used:"
1520 VTAB 18: HTAB 3: PRINT "Word: ";H$
1530 RETURN
1540 REM ENTER LETTER
1550 VTAB 6: HTAB 14 + 2 * N: FLASH : PRINT
7 - N: NORMAL
1560 VTAB 20: HTAB 3: PRINT SPC( 36)
1570 VTAB 20: HTAB 12: PRINT "Press any le
tter "; CHR$ (7);
1580 GET L$
1590 A = ASC (L$)
1600 IF A > 90 THEN A = A - 32
1610 IF A < 65 OR A > 90 THEN 1570
1620 L$ = CHR$ (A)

```

Brain teasers

```
1630 REM MAKE SURE LETTER IS NOT ALREADY
    IN WORD
1640 GOSUB 1720
1650 IF CK$ = "USED" THEN 1560
1660 REM SEARCH FOR LETTER
1670 GOSUB 1810
1680 REM BLOCK OUT NUMBER
1690 IF SHOT$ = "MISS" THEN GOSUB 1910
1700 VTAB 18: HTAB 9: PRINT H$
1710 RETURN
1720 REM MAKE SURE LETTER IS NOT ALREADY
    IN WORD
1730 CK$ = ""
1740 FOR I = 1 TO L
1750 IF LEFT$ ( RIGHT$ (H$,L - I + 1),1) =
    L$ THEN CK$ = "USED"
1760 NEXT
1770 IF CK$ = "" THEN 1800
1780 VTAB 20: HTAB 6: PRINT "Letter is alr
    eady in word !";
1790 FOR PAUSE = 1 TO 300: SOUND = PEEK ( -
    16336): NEXT
1800 RETURN
1810 REM SEARCH FOR LETTER
1820 HD$ = "": SHOT$ = "MISS"
1830 FOR I = 1 TO L
1840 C1$ = LEFT$ ( RIGHT$ (WD$,L - I + 1),
    1)
1850 C2$ = LEFT$ ( RIGHT$ (H$,L - I + 1),1
    )
1860 IF C1$ = L$ THEN HD$ = HD$ + L$: SHOT$
    = "HIT"
1870 IF C1$ < > L$ THEN HD$ = HD$ + C2$
1880 NEXT
1890 H$ = HD$
1900 RETURN
1910 REM BLOCK OUT NUMBER
1920 REM BAD LETTERS
1930 LTR$ = LTR$ + L$ + ", "
1940 VTAB 15: HTAB 9: PRINT LTR$
1950 REM NUMBER
1960 VTAB 6: HTAB 14 + 2 * N: PRINT CHR$
    (32)
1970 N = N + 1
1980 RETURN
1990 REM GUESS
2000 VTAB 20: HTAB 3: PRINT "Want to guess
    the word (Y/N) ? "; CHR$ (7);
2010 GET S$
2020 IF S$ = "N" OR S$ = "n" THEN 2150
```

```

2030 IF S$ < > "Y" AND S$ < > "y" THEN 2
000
2040 VTAB 20: HTAB 3: PRINT SPC( 36)
2050 VTAB 20: HTAB 3: INPUT "Guess = ? ";G
UESS$
2060 VTAB 20: HTAB 39: INVERSE : PRINT CHR$
(32): NORMAL
2070 IF GUESS$ = "" THEN 2040
2080 REM CAPS
2090 GOSUB 2160
2100 T$ = "WRONG !"
2110 IF GUESS$ = WD$ THEN T$ = "RIGHT !"
2120 VTAB 20: HTAB 3: PRINT SPC( 36)
2130 VTAB 20: HTAB 17: PRINT T$;
2140 FOR PAUSE = 1 TO 20: PRINT CHR$ (7);
: NEXT
2150 RETURN
2160 REM CAPS
2170 G$ = ""
2180 LG = LEN (GUESS$)
2190 FOR I = 1 TO LG
2200 LF$ = LEFT$ ( RIGHT$ (GUESS$,LG - I +
1),1)
2210 A = ASC (LF$)
2220 IF A > 90 THEN A = A - 32
2230 LF$ = CHR$ (A)
2240 G$ = G$ + LF$
2250 NEXT
2260 GUESS$ = G$
2270 RETURN
2280 REM WINNER
2290 HOME
2300 PRINT TAB( 11)"CONGRATULATIONS !"
2310 PRINT
2320 PRINT "Correct Word: ";: INVERSE : PRINT
WD$: NORMAL
2330 PRINT
2340 PRINT "You guessed it using ";N;" bad
letters."
2350 RETURN
2360 REM LOSER
2370 HOME
2380 PRINT TAB( 12)"MY CONDOLENCES !"
2390 PRINT
2400 PRINT "Correct Word: ";: INVERSE : PRINT
WD$: NORMAL
2410 PRINT
2420 PRINT "Better luck next time !"
2430 RETURN
2440 REM COMPUTER'S VOCABULARY

```

Brain teasers

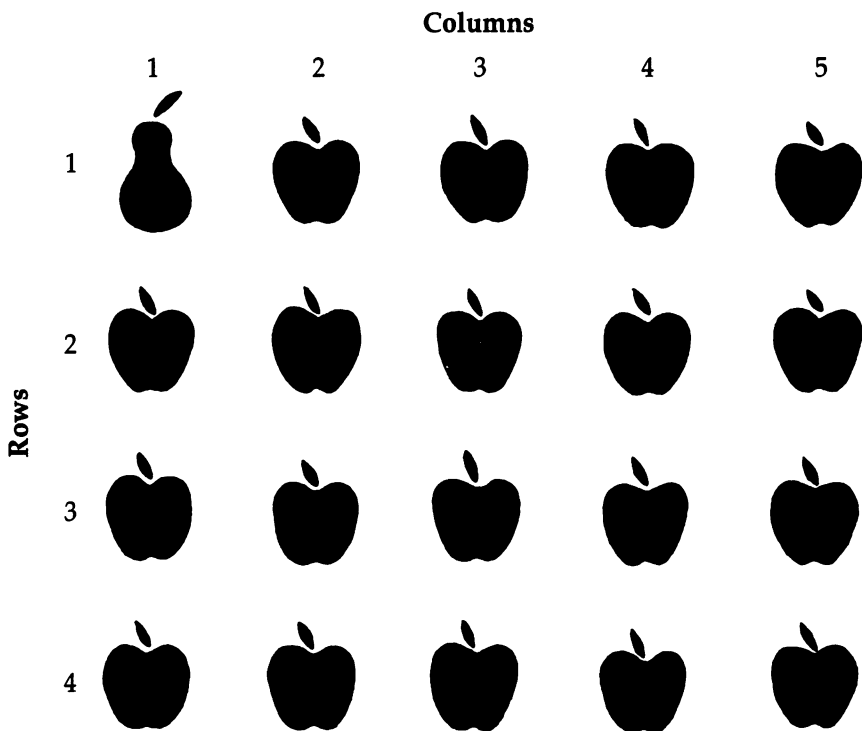
- 2450 DATA ABLE, APPLE, AIRPLANE, ARTILLER
YMAN
- 2460 DATA BOAT, BELLYBUTTON, BAT, BENEDIC
TION
- 2470 DATA CAT, CATERPILLAR, COATS, CATHOL
IC
- 2480 DATA CORN, COSMOPOLITAN, COUNT, COUN
TRY
- 2490 DATA DOG, DRAG, DECEIT, DECOY
- 2500 DATA EEL, EFFICIENCY, EGGHEAD, ELECT
RICIAN
- 2510 DATA FATHER, FEATHER, FELONY, FINICK
Y
- 2520 DATA GOAT, GREAT, GLADIATOR, GNOME
- 2530 DATA HIPPOPOTAMUS, HOLOCAUST, HOLY,
HONEY
- 2540 DATA IDEAL, IMPRISON, ICEMAN, IMMEAS
URABLE
- 2550 DATA JOWL, JAW, JUGGLER, JUVENILE
- 2560 DATA KALEIDOSCOPE, KEEL, KANGAROO, K
ING
- 2570 DATA LABOR, LADLE, LANGUAGE, LISTEN
- 2580 DATA MALLARD, MAP, MANSLAUGHTER, MER
CENARY
- 2590 DATA NIGHT, NICKEL, NUCLEAR, NUMBER
- 2600 DATA OIL, OPPORTUNITY, OUTSTANDING,
OZONE
- 2610 DATA PACK, PAINTER, PENGUIN, POLYNOM
IAL
- 2620 DATA QUACK, QUAGMIRE, QUARTER, QUEAS
Y
- 2630 DATA RUFFLE, RANDOMIZE, REFINED, RHE
UMATIC
- 2640 DATA STASH, SNIFFLE, SWEEPSTAKES, SY
NAGOGUE
- 2650 DATA TATOO, TEMPESTUOUS, THIRSTY, TU
NA
- 2660 DATA UNCOUTH, UNCLE, UPON, USHER
- 2670 DATA VERB, VAT, VICTORY, VOLUNTEER
- 2680 DATA WALRUS, WART, WHITTLE, WORKABLE
- 2690 DATA XYLOPHONE, IMAGINATION, ILLUMIN
ATION, HARMONICA
- 2700 DATA YAM, YEOMANLY, YOGURT, YOUTHFUL
- 2710 DATA ZEBRA, ZESTFUL, ZODIAC, ZOOLOGI
CAL

Chomp

No, this isn't another version of *Pac-Man*. It's a quick game of strategy and planning where you try to make your opponent eat the only pear in a group of apples.

Saving a pear for a friend is usually the sign of thoughtfulness. But not in "Chomp," where you're trying to force the other player to eat a bitter pear. Most of the orchard is filled with delicious apples. In the upper-left corner, however, there's a small pear (see Figure 1). Make your opponent bite that pear and you win this game of thought.

Figure 1. The Orchard



Apples and a Pear

After you've typed and saved Chomp, run it. You'll see a short instruction screen and will be asked if you are using a

color monitor, and then the number of players, which can range from two to five. You will be asked to enter each person's name, and another display then appears.

The computer sets the size of the orchard. From five to seven columns and rows are allowed.

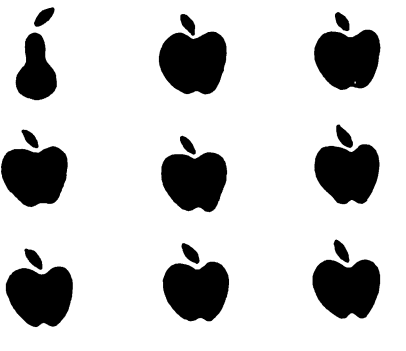
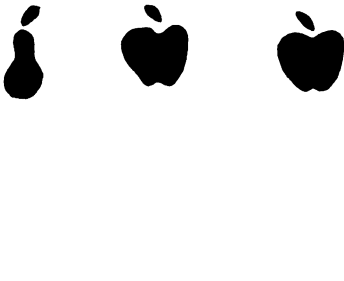
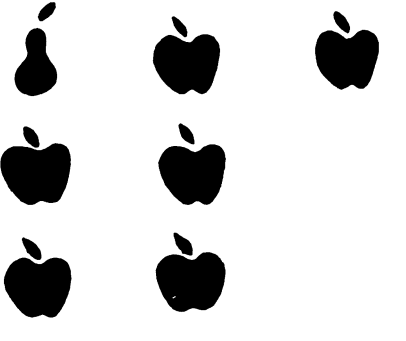

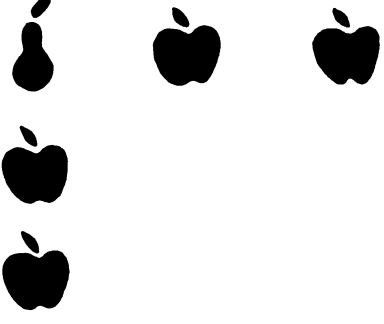
Strategic Eating

You begin eating apples from the lower-right corner, and head inward, either north or west. The number of apples eaten and their locations are determined by the row and column numbers that you enter. All the apples below and to the right of the location you choose are eaten, including that apple. In Figure 1, for instance, entering row 3 and column 1 (for third row, first column) would consume the entire third and fourth rows. Entering 2,1 would eat the second row and the rows below it. Choosing the apple at 2,2 would eliminate the four rightmost apples in rows 2, 3, and 4. You must eat from the lower right toward the north and west. There's no alternative to this direction. Take a look at Figure 2 for an example of a two-player game.

This game originally appeared in the mathematical games section of *Scientific American* in the late 1960s. This version differs slightly in execution but is similar in spirit.

A winning strategy does exist for the two-player case. See if you can discover it. For more general cases, however, a watertight solution has not been figured out, at least to the best of our knowledge. But that's what makes Chomp so fascinating.

Figure 2. Sample Game

<p>Initial Position</p> 	<p>3. Player A enters: Row = 2 & Col = 1</p> 
<p>1. Player A enters: Row = 2 & Col = 3</p> 	<p>4. Player B responds: Row = 1 & Col = 2</p> 
<p>2. Player B responds: Row = 2 & Col = 2</p> 	<p>5. Player A is forced to eat the pear</p>

Chomp

```

100 REM CHOMP
110 REM INITIALIZE
120 GOSUB 210
130 REM PLAY GAME
140 GOSUB 550
150 REM PLAY AGAIN
160 VTAB 23: HTAB 11: PRINT "Play again (Y
/N) ? "; CHR$ (7);
170 GET S$
180 IF S$ = "Y" OR S$ = "y" THEN 140
190 IF S$ < > "N" AND S$ < > "n" THEN 16
0
200 END
210 REM INITIALIZE
220 REM TITLE
230 GOSUB 290
240 REM APPLE & PEAR POINTS
250 GOSUB 360
260 REM INSTRUCTIONS
270 GOSUB 440
280 RETURN
290 REM TITLE
300 PRINT CHR$ (21): TEXT : HOME
310 K = 18
320 DIM NC(7),A1(K),A2(K),P1(K),P2(K)
330 VTAB 12: HTAB 18: PRINT "CHOMP
340 FOR PAUSE = 1 TO 2000: NEXT
350 RETURN
360 REM POINTS
370 REM APPLE
380 DATA 9,10,8,9,7,9,7,8,3,11,2,12,1,13,
1,13,0,14,0,14,0,14,0,14,1,13,1,1
3,2,12,3,11,4,10
390 FOR I = 1 TO K: READ A1(I),A2(I): NEXT

400 REM PEAR
410 DATA 4,5,5,6,7,7,7,7,5,9,4,10,4,10,4,
10,4,10,4,10,3,11,3,11,2,12,2,12,2,12,
2,12,3,11,4,10
420 FOR I = 1 TO K: READ P1(I),P2(I): NEXT

430 RETURN
440 REM INSTRUCTIONS
450 HOME
460 PRINT "The object of CHOMP is to force
your
470 PRINT "opponent to eat the lone, littl
e pear

```

```

480 PRINT "in an orchard full of apples.
490 VTAB 8: HTAB 1: PRINT "Are you using a
    color monitor (Y/N) ? "; CHR$ (7);
500 GET S$
510 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 490
520 C1 = 3:C2 = 3
530 IF S$ = "Y" OR S$ = "y" THEN C1 = 5:C2
    = 6
540 RETURN
550 REM GAME
560 REM PLAYERS
570 GOSUB 680
580 REM INITIAL POSITION
590 GOSUB 840
600 REM BITE
610 GOSUB 1530
620 REM CHECK FOR END OF GAME
630 IF NC(1) = 0 THEN GOSUB 1920
640 PLAYER = PLAYER + 1
650 IF PLAYER = N + 1 THEN PLAYER = 1
660 IF NC(1) < > 0 THEN 610
670 RETURN
680 REM PLAYERS
690 HOME
700 VTAB 1: HTAB 1: PRINT "How many player
    s are there (2 to 5) ?"; CHR$ (7);
710 GET S$
720 N = VAL (S$)
730 IF N < 2 OR N > 5 THEN 700
740 REM NAMES
750 VTAB 3: HTAB 1: PRINT "Please enter yo
    ur names. Up to 10
760 PRINT "characters are allowed.
770 FOR I = 1 TO N
780 VTAB 6: HTAB 16: PRINT SPC( 20); CHR$
    (7)
790 VTAB 6: HTAB 1: PRINT "Name No. ";I;: INPUT
    " = ? ";N$
800 IF N$ = "" OR LEN (N$) > 10 THEN 780
810 N$(I) = N$
820 NEXT
830 RETURN
840 REM INITIAL POSITION
850 REM ORCHARD SIZE
860 GOSUB 940
870 REM LABELING
880 GOSUB 980
890 REM ORCHARD
900 GOSUB 1170

```

```

910 REM COUNTERS
920 GOSUB 1430
930 RETURN
940 REM SIZE
950 ROWS = INT ( RND (1) * 4) + 4
960 COLS = INT ( RND (1) * 4) + 4
970 RETURN
980 REM LABELING
990 HOME
1000 HGR : HCOLOR= 0: HPLLOT 1,1: CALL 6245
    4
1010 HCOLOR= 3
1020 REM COLUMNS
1030 FOR I = 1 TO COLS
1040 X = 30 * I + 5:Y = 10
1050 ON I GOSUB 2000,2030,2080,2120,2160,2
    200,2250
1060 NEXT
1070 REM ROWS
1080 FOR I = 1 TO ROWS
1090 X = 15:Y = 20 * I + 13
1100 ON I GOSUB 2000,2030,2080,2120,2160,2
    200,2250
1110 NEXT
1120 FOR I = 1 TO 3
1130 X = 1:Y = 10 * I + 65
1140 ON I GOSUB 2290,2330,2360
1150 NEXT
1160 RETURN
1170 REM ORCHARD
1180 HCOLOR= C1
1190 REM APPLES
1200 FOR I = 1 TO COLS
1210 FOR J = 1 TO ROWS
1220 X = 30 * I:Y = 20 * J
1230 FOR L = 1 TO K
1240 HPLLOT X + A1(L),Y TO X + A2(L),Y
1250 Y = Y + 1
1260 NEXT L,J,I
1270 REM PEAR
1280 REM DARKEN BLOCK
1290 X = 1:Y = 1: GOSUB 1360
1300 HCOLOR= C2:Y = 20
1310 FOR I = 1 TO K
1320 HPLLOT 30 + P1(I),Y TO 30 + P2(I),Y
1330 Y = Y + 1
1340 NEXT
1350 RETURN
1360 REM DARKEN BLOCK
1370 HCOLOR= 0

```


Brain teasers

```
1380 X0 = 30 * X:Y0 = 20 * Y
1390 FOR L = X0 TO X0 + 15
1400 HPLLOT L,Y0 TO L,Y0 + 18
1410 NEXT L
1420 RETURN
1430 REM COUNTERS
1440 PLAYER = 1
1450 REM # OF COLUMNS IN EACH ROW
1460 FOR I = 1 TO 7
1470 NC(I) = 0
1480 IF I < = ROWS THEN NC(I) = COLS
1490 NEXT
1500 REM TOTAL # OF ROWS
1510 TR = ROWS
1520 RETURN
1530 REM BITE
1540 REM ENTER ROW
1550 GOSUB 1630
1560 REM ENTER COLUMN
1570 GOSUB 1730
1580 REM TAKE BITE
1590 GOSUB 1800
1600 REM UPDATE COUNTERS
1610 GOSUB 1860
1620 RETURN
1630 REM ROW
1640 VTAB 21: HTAB 13: PRINT SPC( 20)
1650 VTAB 23: HTAB 1: PRINT SPC( 40)
1660 VTAB 21: HTAB 1: PRINT "Please bite "
;N$(PLAYER);","
1670 VTAB 23: HTAB 1: PRINT "Row = ? "; CHR$
(7);
1680 GET S$
1690 R = VAL (S$)
1700 IF R < 1 OR R > TR THEN 1670
1710 PRINT R
1720 RETURN
1730 REM COLUMN
1740 VTAB 23: HTAB 15: PRINT "Column = ? "
; CHR$ (7);
1750 GET S$
1760 C = VAL (S$)
1770 IF C < 1 OR C > NC(R) THEN 1740
1780 PRINT C
1790 RETURN
1800 REM TAKE BITE
1810 FOR Y = R TO TR
1820 FOR X = C TO NC(Y)
1830 IF C < = NC(Y) THEN GOSUB 1360
1840 NEXT X,Y
```

```

1850 RETURN
1860 REM COUNTERS
1870 FOR I = R TO TR
1880 IF NC(I) >= C THEN NC(I) = C - 1
1890 NEXT
1900 IF C = 1 THEN TR = R - 1
1910 RETURN
1920 REM GAME OVER
1930 TEXT
1940 HOME
1950 PRINT "So sorry, ";N$(PLAYER);"."
1960 PRINT
1970 PRINT "You ate the little pear and n
ow have
1980 PRINT "an upset tummy !"
1990 RETURN
2000 REM 1
2010 HPLLOT X + 1,Y TO X + 3,Y: HPLLOT X + 2
,Y TO X + 2,Y - 6: HPLLOT X + 1,Y - 5
2020 RETURN
2030 REM 2
2040 HPLLOT X,Y TO X + 4,Y: HPLLOT X,Y - 1: HPLLOT
X + 1,Y - 2
2050 HPLLOT X + 2,Y - 3 TO X + 3,Y - 3: HPLLOT
X + 4,Y - 4 TO X + 4,Y - 5
2060 HPLLOT X + 1,Y - 6 TO X + 3,Y - 6: HPLLOT
X,Y - 5
2070 RETURN
2080 REM 3
2090 HPLLOT X,Y - 6 TO X + 4,Y - 6: HPLLOT X
+ 1,Y TO X + 3,Y: HPLLOT X + 2,Y - 3 TO
X + 3,Y - 3
2100 HPLLOT X,Y - 1: HPLLOT X + 4,Y - 1 TO X
+ 4,Y - 2: HPLLOT X + 3,Y - 4: HPLLOT X
+ 4,Y - 5
2110 RETURN
2120 REM 4
2130 HPLLOT X + 3,Y TO X + 3,Y - 6: HPLLOT X
,Y - 2 TO X + 4,Y - 2
2140 HPLLOT X,Y - 3: HPLLOT X + 1,Y - 4: HPLLOT
X + 2,Y - 5
2150 RETURN
2160 REM 5
2170 HPLLOT X,Y - 6 TO X + 4,Y - 6: HPLLOT X
,Y - 4 TO X + 3,Y - 4: HPLLOT X + 4,Y -
1 TO X + 4,Y - 3: HPLLOT X + 1,Y TO X +
3,Y
2180 HPLLOT X,Y - 5: HPLLOT X,Y - 1
2190 RETURN
2200 REM 6

```

```

2210 H PLOT X + 1,Y TO X + 3,Y: H PLOT X,Y -
    3 TO X + 3,Y - 3: H PLOT X + 2,Y - 6 TO
    X + 4,Y - 6
2220 H PLOT X + 1,Y - 5: H PLOT X,Y - 4
2230 H PLOT X,Y - 1 TO X,Y - 2: H PLOT X + 4
    ,Y - 1 TO X + 4,Y - 2
2240 RETURN
2250 REM 7
2260 H PLOT X + 1,Y TO X + 1,Y - 2: H PLOT X
    ,Y - 6 TO X + 4,Y - 6
2270 H PLOT X + 2,Y - 3: H PLOT X + 3,Y - 4:
    H PLOT X + 4,Y - 5
2280 RETURN
2290 REM R
2300 H PLOT X,Y TO X,Y - 6: H PLOT X + 1,Y -
    6 TO X + 3,Y - 6: H PLOT X + 1,Y - 3 TO
    X + 3,Y - 3: H PLOT X + 4,Y - 4 TO X +
    4,Y - 5
2310 H PLOT X + 2,Y - 2: H PLOT X + 3,Y - 1:
    H PLOT X + 4,Y
2320 RETURN
2330 REM 0
2340 H PLOT X,Y - 1 TO X,Y - 5: H PLOT X + 4
    ,Y - 1 TO X + 4,Y - 5: H PLOT X + 1,Y TO
    X + 3,Y: H PLOT X + 1,Y - 6 TO X + 3,Y -
    6
2350 RETURN
2360 REM W
2370 H PLOT X,Y TO X,Y - 6: H PLOT X + 4,Y TO
    X + 4,Y - 6
2380 H PLOT X + 1,Y - 1: H PLOT X + 3,Y - 1:
    H PLOT X + 2,Y - 2: H PLOT X + 2,Y - 3
2390 RETURN

```

Math Man

Original Game Concept by Andy Hayes

Test your mathematical abilities in this timed game of addition, subtraction, multiplication, and division.

The key to success at "Math Man" is being able to add, subtract, multiply, and divide quickly and accurately. This fast-paced game asks for answers in two-number addition, subtraction, multiplication, and division problems. To keep you on your toes, the Math Man sets a clock to time you.

Grade School and Beyond

Type in, save, and run Math Man. As in the first brainteaser in this chapter, "Guess a Word," you need to enter a number from 1 to 1000. Make sure you type in a different number for each session of Math Man. If you use the same number over and over, the order of the problems will remain the same. The easiest way to insure a different set of problems each time is just to randomly type some numbers on the keyboard.

After you enter a number, you'll choose whether you want to play against the computer or against human competition. Enter all players' names; if you're playing alone against the computer, enter just your own name.

Three difficulty levels are available: easy, medium, and hard. Each round consists of ten problems. The easy level is more appropriate for beginners, since the computer always chooses integers between 1 and 10 to keep things simple. A typical problem might be $7 + 10 = ?$. The medium level is a bit more difficult, with numbers up to 100 appearing. And the hard level should be attempted only if you think you're a top-notch math wizard. You might see something like $338 * 524 = ?$ in this level, since integers from 1 to 1000 can appear.

There's a time limit for each problem. You're allowed 15 seconds for addition and subtraction, 30 seconds for multiplication, and 45 seconds for division. If you don't enter an answer before the time is up, it's counted as incorrect.

If you want, you can reset the number of seconds allowed for an answer. The minimum time allowed is 10 seconds and the maximum is 60. Set your own time limits before you begin play.

Random Skill Level

Math Man randomly generates the two integers, as well as the required operation (+, -, *, or /). Each occurs equally often; any streaks of simpler problems will be offset by more difficult ones.

Don't lose heart; Math Man does a couple of things to ease your mathematical burden. First, division problems can be rounded to two decimal places. After all, it might take all day evaluating something like $22/7$. If you see a problem such as $10/3$, for example, type in 3.33. If you entered only 3.3, it would count as incorrect, since Math Man considers *two* decimal places as significant. Second, Math Man sounds a warning buzzer when you have only five seconds left to solve the problem. This is a handy feature when you're madly scratching out your answer on a piece of paper and not looking at the screen.

Math Man allows up to ten players. The person with the most correct answers at the end of the game wins. In the case of ties, the player using the fewest seconds is champion.

Math Man

```

100 REM MATH MAN
110 REM INITIALIZE
120 GOSUB 220
130 REM GAME
140 GOSUB 610
150 REM DISPLAY RESULT
160 GOSUB 3100
170 VTAB 23: HTAB 11: PRINT "Play again (Y
    /N) ?"; CHR$ (7);
180 GET S$
190 IF S$ = "Y" OR S$ = "y" THEN 140
200 IF S$ < > "N" AND S$ < > "n" THEN 17
    0
210 END
220 REM INITIALIZE
230 REM TITLE
240 GOSUB 300
250 REM INSTRUCTIONS
260 GOSUB 350
270 REM VALUES
280 GOSUB 550
290 RETURN
300 REM TITLE
310 PRINT CHR$ (21): TEXT : HOME
320 VTAB 13: HTAB 15: PRINT "Math Man
    
```

```

330 FOR PAUSE = 1 TO 2000: NEXT
340 RETURN
350 REM INSTRUCTIONS
360 HOME
370 PRINT "Math Man tests your ability to
quickly
380 PRINT "add, subtract, multiply, and di
vide.
390 PRINT
400 PRINT "Three levels of difficulty are
410 PRINT "available, with 10 problems in
each.
420 PRINT
430 PRINT "Please round division answers t
o two
440 PRINT "decimal places.
450 PRINT
460 PRINT "And please enter a number so th
at I can"
470 PRINT "choose problems randomly.
480 VTAB 13: HTAB 22: PRINT SPC( 20); CHR$
(7);
490 VTAB 13: HTAB 1: INVERSE : PRINT "SEED
(1 TO 1000)";: NORMAL
500 INPUT " = ? ";S$
510 V = VAL (S$)
520 IF V < 1 OR V > 1000 THEN 480
530 Z = RND ( - V)
540 RETURN
550 REM SECONDS ALLOWED FOR +,-,*, & /
560 DATA 15,15,30,45
570 FOR I = 1 TO 4
580 READ S(I)
590 NEXT
600 RETURN
610 REM GAME
620 REM CHOOSE SKILL LEVEL
630 GOSUB 730
640 REM DISPLAY SECONDS ALLOWED
650 GOSUB 890
660 REM ENTER PLAYERS
670 GOSUB 1250
680 REM SET INITIAL VALUES
690 GOSUB 1430
700 REM PLAY
710 GOSUB 1490
720 RETURN
730 REM SKILL LEVEL
740 HOME

```

```

750 VTAB 7: HTAB 9: PRINT "Difficulty of P
    roblems
760 INVERSE
770 FOR I = 1 TO 3
780 VTAB I * 2 + 7: HTAB 10: PRINT I
790 NEXT
800 NORMAL
810 VTAB 9: HTAB 12: PRINT "Easy
820 VTAB 11: HTAB 12: PRINT "Medium
830 VTAB 13: HTAB 12: PRINT "Hard
840 VTAB 15: HTAB 9: PRINT "Choice = ? "; CHR$
    (7);
850 GET S$
860 D = VAL (S$)
870 IF D < 1 OR D > 3 THEN 840
880 RETURN
890 REM SECONDS ALLOWED
900 HOME
910 VTAB 4: HTAB 29: PRINT "Seconds"
920 VTAB 5: HTAB 9: PRINT "Operation"; TAB(
    29)"Allowed"
930 INVERSE
940 FOR I = 1 TO 4
950 VTAB 2 * I + 5: HTAB 5: PRINT I
960 NEXT
970 NORMAL
980 VTAB 7: HTAB 7: PRINT "Addition"; TAB(
    32)S(1)
990 VTAB 9: HTAB 7: PRINT "Subtraction"; TAB(
    32)S(2)
1000 VTAB 11: HTAB 7: PRINT "Multiplicatio
    n"; TAB( 32)S(3)
1010 VTAB 13: HTAB 7: PRINT "Division"; TAB(
    32)S(4)
1020 REM MAKE CHANGES
1030 VTAB 16: HTAB 13: PRINT "Changes (Y/N
    ) ? "; CHR$ (7);
1040 GET S$
1050 IF S$ = "Y" OR S$ = "y" THEN GOSUB 1
    080: GOTO 1030
1060 IF S$ < > "N" AND S$ < > "n" THEN 1
    030
1070 RETURN
1080 REM CHANGES
1090 VTAB 18: HTAB 13: PRINT "Number (1 to
1100 GET S$
1110 V = VAL (S$)
1120 IF V < 1 OR V > 4 THEN 1090
1130 PRINT V

```

```

1140 VTAB 20: HTAB 25: PRINT SPC( 15); CHR$
    (7)
1150 VTAB 20: HTAB 13: INPUT "Seconds = ?
    ";S$
1160 S = INT ( VAL (S$))
1170 VTAB 23: HTAB 5: PRINT SPC( 30)
1180 IF S < 10 THEN VTAB 23: HTAB 8: PRINT
    "At least 10 are needed !": GOTO 1140
1190 IF S > 60 THEN VTAB 23: HTAB 5: PRINT
    "No more than 60 are allowed !": GOTO
    1140
1200 S(V) = S: VTAB 2 * V + 5: HTAB 32: PRINT
    S
1210 VTAB 18: HTAB 13: PRINT SPC( 22)
1220 VTAB 20: HTAB 13: PRINT SPC( 14)
1230 VTAB 23: HTAB 5: PRINT SPC( 30)
1240 RETURN
1250 REM PLAYERS
1260 HOME
1270 VTAB 1: HTAB 1: PRINT "How many playe
    rs are there (1 to 9) ? "; CHR$ (7);
1280 GET S$
1290 N = VAL (S$)
1300 IF N < 1 OR N > 9 THEN 1270
1310 N$ = "names"
1320 IF N = 1 THEN N$ = "name"
1330 VTAB 3: HTAB 1: PRINT "Please enter y
    our ";N$; ". Up to 15
1340 PRINT "characters are allowed.
1350 FOR I = 1 TO N
1360 VTAB 6: HTAB 18: PRINT SPC( 25); CHR$
    (7)
1370 VTAB 6: HTAB 1: PRINT "Player No. ";I
    ;
1380 INPUT " = ? ";N$
1390 IF N$ = "" OR LEN (N$) > 15 THEN 136
    0
1400 N$(I) = N$
1410 NEXT
1420 RETURN
1430 REM INITIAL VALUES
1440 FOR I = 1 TO N
1450 SC(I) = 0
1460 RG(I) = 0
1470 NEXT
1480 RETURN
1490 REM PLAY
1500 FOR I = 1 TO N
1510 HOME
1520 PRINT "Your turn ";N$(I); ". "

```



```

1530 VTAB 3: HTAB 14: INVERSE : PRINT "PRE
    SS ANY KEY";: NORMAL : PRINT " "; CHR$
    (7);
1540 GET S$
1550 REM DRAW BOARD
1560 GOSUB 1680
1570 FOR J = 1 TO 10
1580 REM POSE PROBLEM
1590 GOSUB 1950
1600 REM GET ANSWER
1610 GOSUB 2190
1620 REM EVALUATE IT
1630 GOSUB 2720
1640 NEXT J
1650 FOR PAUSE = 1 TO 2500: NEXT PAUSE
1660 NEXT I
1670 RETURN
1680 REM BOARD
1690 REM BOX
1700 GOSUB 1740
1710 REM LABEL
1720 GOSUB 1880
1730 RETURN
1740 REM BOX
1750 HOME : INVERSE :BK$ = CHR$ (32)
1760 FOR L = 1 TO 39
1770 VTAB 1: HTAB L: PRINT BK$
1780 VTAB 2: HTAB L: PRINT BK$
1790 VTAB 3: HTAB L: PRINT BK$
1800 VTAB 24: HTAB L: PRINT BK$;
1810 NEXT L
1820 FOR L = 4 TO 23
1830 VTAB L: HTAB 1: PRINT BK$
1840 VTAB L: HTAB 39: PRINT BK$
1850 NEXT L
1860 NORMAL
1870 RETURN
1880 REM LABEL
1890 INVERSE : VTAB 11: HTAB 11: PRINT "PR
    OBLEM:"
1900 VTAB 13: HTAB 12: PRINT "ANSWER:": NORMAL

1910 VTAB 2: HTAB 15: PRINT " MATH MAN "
1920 VTAB 5: HTAB 28: PRINT "Number"
1930 VTAB 6: HTAB 28: PRINT "Right = 0"
1940 RETURN
1950 REM POSE PROBLEM
1960 VTAB 22: HTAB 6: PRINT SPC( 30)
1970 VTAB 22: HTAB 6: PRINT "Press ";: INVERSE
    : PRINT "RETURN";: NORMAL : PRINT " fo
    r Problem# ";J;" "; CHR$ (7);

```

```

1980 GET S$
1990 IF ASC (S$) < > 13 THEN 1970
2000 REM CLEAR AREA
2010 GOSUB 2140
2020 REM NUMBERS
2030 DEF FN T(V) = INT (V * RND (1) + 1
)
2040 X = FN T(10 ^ D):Y = FN T(10 ^ D)
2050 REM OPERATION
2060 Z = INT (4 * RND (1) + 1)
2070 IF Z = 1 THEN ANW = X + Y:S$ = " + "
2080 IF Z = 2 THEN ANW = X - Y:S$ = " - "
2090 IF Z = 3 THEN ANW = X * Y:S$ = " * "
2100 IF Z = 4 THEN ANW = X / Y:S$ = " / "
2110 P$ = STR$ (X) + S$ + STR$ (Y)
2120 VTAB 11: HTAB 20: PRINT P$
2130 RETURN
2140 REM CLEAR AREA
2150 VTAB 11: HTAB 20: PRINT SPC( 15)
2160 VTAB 13: HTAB 20: PRINT SPC( 15)
2170 VTAB 15: HTAB 6: PRINT SPC( 30)
2180 RETURN
2190 REM GET ANSWER
2200 REM INITIAL VALUES
2210 GOSUB 2250
2220 REM ANSWER
2230 GOSUB 2310
2240 RETURN
2250 REM VALUES
2260 VTAB 22: HTAB 6: PRINT SPC( 28)
2270 VTAB 22: HTAB 11: INVERSE : PRINT "SE
CONDS": NORMAL
2280 TICS = S(Z)
2290 A$ = "":C = 20
2300 RETURN
2310 REM ANSWER
2320 MOVE$ = ""
2330 REM TIME
2340 GOSUB 2420
2350 FOR L = 1 TO 4
2360 P = PEEK ( - 16384)
2370 NEXT L
2380 IF TICS < = 5 THEN FOR L = 1 TO 2:S
OUND = PEEK ( - 16336): NEXT L
2390 IF P > = 128 THEN GOSUB 2470
2400 IF TICS > 0.13 AND MOVE$ < > "OVER" THEN
2340
2410 RETURN
2420 REM TIME
2430 IF TICS < 0 THEN TICS = 0

```

```

2440 VTAB 22: HTAB 20: PRINT SPC( 2);: HTAB
    20: PRINT INT (TICS)
2450 TICS = TICS - 0.12
2460 RETURN
2470 REM RESPONSE
2480 MOVE$ = ""
2490 A = P - 128
2500 IF A < > 8 AND A < > 13 AND A < >
    45 AND A < > 46 AND (A < 48 OR A > 57
    ) THEN PRINT CHR$ (7);:MOVE$ = "BAD"

2510 IF A = 13 THEN MOVE$ = "OVER"
2520 IF A = 8 THEN GOSUB 2570
2530 IF MOVE$ = "" THEN GOSUB 2660
2540 POKE - 16368,0
2550 TICS = TICS - 0.12
2560 RETURN
2570 REM SUBTRACT DIGIT
2580 IF A$ = "" THEN PRINT CHR$ (7);: GOTO
    2640
2590 K = LEN (A$)
2600 IF K = 1 THEN A$ = ""
2610 IF K > 1 THEN A$ = LEFT$ (A$,K - 1)
2620 C = C - 1
2630 VTAB 13: HTAB C: PRINT " "
2640 MOVE$ = "OKAY"
2650 RETURN
2660 REM ADD DIGIT
2670 VTAB 13: HTAB C: PRINT CHR$ (A)
2680 A$ = A$ + CHR$ (A)
2690 C = C + 1
2700 MOVE$ = "OKAY"
2710 RETURN
2720 REM EVALUATE ANSWER
2730 A = VAL (A$):R$ = "WRONG !"
2740 REM +,-, OR *
2750 IF Z < > 4 THEN GOSUB 2850
2760 REM /
2770 IF Z = 4 THEN GOSUB 2880
2780 REM FLASH RESULT
2790 GOSUB 2920
2800 REM DISPLAY RIGHT ANSWER
2810 IF R$ = "WRONG !" THEN GOSUB 2970
2820 REM ADD TO TOTALS
2830 GOSUB 3030
2840 RETURN
2850 REM +,-, OR *
2860 IF A = ANW THEN R$ = "RIGHT !"
2870 RETURN
2880 REM DIVISION

```

```

2890 DEF FN DIV(X) = INT ((X + .005) * 1
    00)
2900 IF FN DIV(A) = FN DIV(ANW) THEN R$ =
    "RIGHT !"
2910 RETURN
2920 REM RESULT
2930 VTAB 22: HTAB 11: PRINT SPC( 11)
2940 VTAB 22: HTAB 17: FLASH : PRINT R$;: NORMAL
2950 FOR PAUSE = 1 TO 20: PRINT CHR$ (7);
    : NEXT PAUSE
2960 RETURN
2970 REM RIGHT ANSWER
2980 VTAB 15: HTAB 6: INVERSE : PRINT "RIG
    HT ANSWER:":; NORMAL :
2990 PRINT " ";
3000 IF Z = 4 THEN ANW = INT ((ANW + .005
    ) * 100) / 100
3010 PRINT ANW
3020 RETURN
3030 REM ADD TO TOTALS
3040 REM TOTAL SECONDS
3050 SC(I) = SC(I) + S(Z) - TICS
3060 REM NUMBER RIGHT
3070 IF R$ = "RIGHT !" THEN RG(I) = RG(I) +
    1
3080 VTAB 6: HTAB 36: PRINT RG(I)
3090 RETURN
3100 REM FINAL SCORE
3110 HOME
3120 PRINT TAB( 15)"FINAL SCORE"
3130 INVERSE : VTAB 3: HTAB 18: PRINT "NUM
    BER":; HTAB 29: PRINT "SECONDS"
3140 VTAB 4: HTAB 1: PRINT "PLAYER":; HTAB
    18: PRINT "RIGHT ":; HTAB 29: PRINT "
    USED ": NORMAL
3150 PRINT
3160 FOR I = 1 TO N
3170 SC(I) = INT (SC(I))
3180 PRINT N$(I); TAB( 22 - LEN ( STR$ (R
    G(I))))RG(I); TAB( 34 - LEN ( STR$ (S
    C(I))))SC(I)
3190 NEXT
3200 RETURN

```

Matches

In this two-player game of logic, you take turns picking up matches, trying to remove the single lighted match at the end.

It's not as simple as it looks. In fact, it's an ideal brainteaser. You're trying to outwit your opponent, human or computer, and get to the single lighted match.

Type in and save "Matches." Run it, and you'll see a short instruction screen. Next, indicate whether you want to play against the computer or a human opponent. Enter the information as requested. After this you will see the main display screen.

One Lighted Match

The computer displays between 7 and 17 matches in a row, as the example in Figure 1 shows.

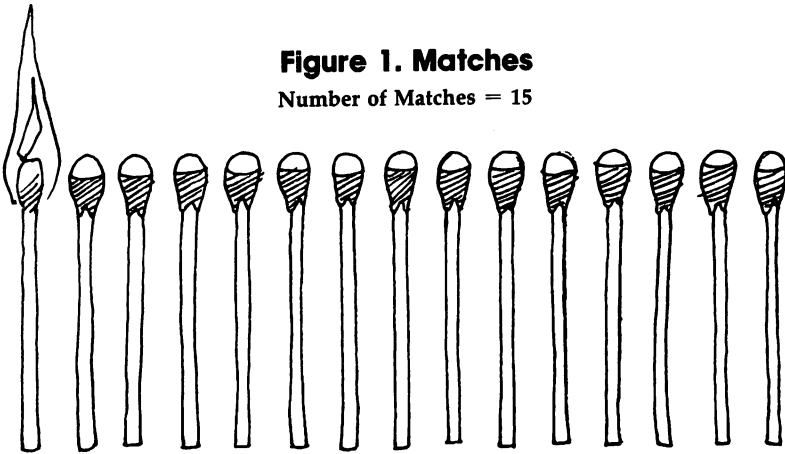


Figure 1. Matches

Number of Matches = 15

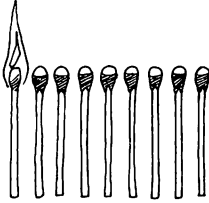
Only the leftmost match is lighted. You and your opponent (whether the computer or another player) take turns removing matches, starting at the right. You must take from one to three matches each term. The player who picks up the flaming match wins.

The concept is simple. But the execution isn't as easy. Take a look at Figure 2, which illustrates a sample game.

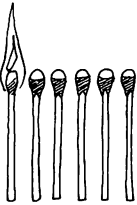
Figure 2. Sample Matches

Initial Position

Number of Matches = 9



**1. First player takes 3 matches.
6 are left:**



**2. Second player takes 2 matches.
4 are left:**



**3. First player takes 1 match.
3 are left:**



**4. Second player takes 3
matches and wins.**

Matching Your Opponent's Skill

Be forewarned that the Apple never makes a mistake. Well, almost never. It can be beaten, but it's difficult. The computer graciously allows you to go first. You'll probably need it.

A good strategy in this game is to think backwards. That is, try to determine which position you'd like to be in with only three or four matches left. Then work backwards to get there.

Watch the computer's moves for several games. See if you can discover the winning pattern. You can really impress your opponents after you've discovered the secret.

Matches

```

100 REM MATCHES
110 REM INITIALIZE
120 GOSUB 210
130 REM PLAY GAME
140 GOSUB 580
150 REM PLAY AGAIN
160 VTAB 23: HTAB 9: PRINT "Play again (Y/
    N) ? "; CHR$ (7);
170 GET S$
180 IF S$ = "Y" OR S$ = "y" THEN 140
190 IF S$ < > "N" AND S$ < > "n" THEN 16
    0
200 END
210 REM INITIALIZE
220 REM TITLE
230 GOSUB 290
240 REM INSTRUCTIONS
250 GOSUB 340
260 REM POINTS FOR FLAME
270 GOSUB 510
280 RETURN
290 REM TITLE
300 PRINT CHR$ (21): TEXT : HOME
310 VTAB 12: HTAB 16: PRINT "Matches
320 FOR PAUSE = 1 TO 2000: NEXT
330 RETURN
340 REM INSTRUCTIONS
350 HOME
360 PRINT "I'm about to display between 7
    and 17
370 PRINT "matches in a row.
380 PRINT
390 PRINT "Only the first one is lit, howe
    ver,
400 PRINT "and your objective is to pick i
    t up.
410 PRINT
420 PRINT "You and your opponent start at
    the end
430 PRINT "of the row and take away 1, 2,
    or 3
440 PRINT "sticks at a time.
450 VTAB 11: HTAB 1: PRINT "Are you using
    a color monitor (Y/N) ? "; CHR$ (7);
460 GET S$
470 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 450
480 C1 = 0: C2 = 3: C3 = 3: C4 = 3

```

```

490 IF S$ = "Y" OR S$ = "y" THEN C1 = 6:C2
    = 0:C3 = 3:C4 = 5
500 RETURN
510 REM FLAME
520 DIM F1(13),F2(13)
530 DATA 61,65,58,68,55,60,51,59,46,58,39
    ,58,34,58,37,58,50,58,52,59,54,60,56,6
    8,60,67
540 FOR I = 1 TO 13
550 READ F1(I),F2(I)
560 NEXT
570 RETURN
580 REM PLAY GAME
590 REM PLAYERS
600 GOSUB 730
610 REM INITIAL POSITION
620 GOSUB 970
630 REM MAKE MOVE
640 GOSUB 1510
650 IF T = 0 THEN WINNER$ = NM$(PLAYER)
660 REM TAKE TURNS
670 PLAYER = PLAYER + 1
680 IF PLAYER = 3 THEN PLAYER = 1
690 IF T > 0 THEN 640
700 REM END OF GAME
710 GOSUB 2250
720 RETURN
730 REM PLAYERS
740 REM SELECT OPPONENT
750 GOSUB 790
760 REM NAMES
770 GOSUB 880
780 RETURN
790 REM OPPONENT
800 HOME
810 PRINT "Two players are needed for Matc
    hes.
820 VTAB 3: HTAB 1: PRINT "Would you like
    to play me (Y/N) ? "; CHR$(7);
830 GET S$
840 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 820
850 N$ = "names":K = 2
860 IF S$ = "Y" OR S$ = "y" THEN NM$(2) =
    "COMPUTER":N$ = "name":K = 1
870 RETURN
880 REM NAMES
890 VTAB 9: HTAB 1: PRINT "Please enter yo
    ur ";N$;". Up to 10"
900 PRINT "characters are allowed.

```



```

910  FOR I = 1 TO K
920  VTAB 12: HTAB 18: PRINT SPC( 30); CHR$(
    7)
930  VTAB 12: HTAB 1: PRINT "Player No. ";I
    ;: INPUT " = ? ";NM$(I)
940  IF NM$(I) = "" OR LEN (NM$(I)) > 10 OR
    NM$(I) = "COMPUTER" THEN 920
950  NEXT
960  RETURN
970  REM INITIAL POSITION
980  REM NUMBER OF STICKS
990  T = 7 + INT ( RND (1) * 11)
1000 REM BOARD
1010 GOSUB 1050
1020 REM COUNTER
1030 PLAYER = 1
1040 RETURN
1050 REM BOARD
1060 HOME
1070 HGR : HCOLOR= C1: HPLOT 1,1: CALL 624
    54
1080 HCOLOR= 0: HPLOT 0,159 TO 279,159
1090 REM STICKS
1100 GOSUB 1170
1110 REM HEADS
1120 GOSUB 1270
1130 REM FLAME
1140 GOSUB 1400
1150 VTAB 21: HTAB 30: INVERSE : PRINT "MA
    TCHES": NORMAL
1160 RETURN
1170 REM STICKS
1180 HCOLOR= C2
1190 X = 20
1200 FOR I = 1 TO T
1210 FOR J = 1 TO 5
1220 HPLOT X + J,70 TO X + J,140
1230 NEXT J
1240 X = X + 15
1250 NEXT I
1260 RETURN
1270 REM HEADS
1280 HCOLOR= C3
1290 X = 21
1300 FOR I = 1 TO T
1310 FOR J = X - 1 TO X + 5
1320 HPLOT J,60 TO J,69
1330 NEXT J
1340 HPLOT X - 2,61 TO X - 2,68
1350 HPLOT X + 6,61 TO X + 6,68

```

```

1360 H PLOT X,59 TO X + 4,59
1370 X = X + 15
1380 NEXT I
1390 RETURN
1400 REM FLAME
1410 HCOLOR= C4
1420 X = 17
1430 FOR I = 1 TO 13
1440 H PLOT X,F1(I) TO X,F2(I)
1450 X = X + 1
1460 NEXT
1470 H PLOT 25,39 TO 25,41
1480 HCOLOR= C1
1490 H PLOT 21,55 TO 21,56: H PLOT 22,47 TO
22,55: H PLOT 23,46 TO 23,55: H PLOT 24,
52 TO 24,54
1500 RETURN
1510 REM MAKE MOVE
1520 V TAB 21: H TAB 38: PRINT SPC( 2)
1530 V TAB 21: H TAB 38: PRINT T
1540 REM CHECK FOR COMPUTER'S MOVE
1550 H$ = ""
1560 IF NM$(PLAYER) = "COMPUTER" THEN GOSUB
1620
1570 REM GET MOVE
1580 GOSUB 1700
1590 REM TAKE STICKS AWAY
1600 GOSUB 2110
1610 RETURN
1620 REM COMPUTER
1630 COUNT = 10:H$ = "Hmmm ..."
1640 REM NUMBER THAT SHOULD BE LEFT
1650 S = INT (T / 4) * 4
1660 REM NUMBER TO TAKE
1670 N = T - S
1680 IF N = 0 THEN N = 1
1690 RETURN
1700 REM GET MOVE
1710 MOVE$ = ""
1720 V TAB 23: H TAB 10: PRINT SPC( 30)
1730 V TAB 23: H TAB 1: PRINT "How many ";NM
$(PLAYER);" ";: FLASH : PRINT "?"; CHR$
(7);: NORMAL
1740 V TAB 23: H TAB 23: PRINT H$;
1750 REM FLAMES 1 & 2
1760 HCOLOR= C4: GOSUB 1990
1770 HCOLOR= C1: GOSUB 2050
1780 REM FLAMES 1 & 2
1790 HCOLOR= C1: GOSUB 1990
1800 HCOLOR= C4: GOSUB 2050

```

```

1810 REM FLAME AGAIN
1820 IF NM$(PLAYER) < > "COMPUTER" AND P <
128 THEN 1760
1830 REM CHECK FOR LEGAL RESPONSE
1840 IF NM$(PLAYER) = "COMPUTER" THEN GOSUB
1880
1850 IF NM$(PLAYER) < > "COMPUTER" THEN GOSUB
1930
1860 IF MOVE$ = "" THEN 1760
1870 RETURN
1880 REM COMPUTER'S MOVE
1890 COUNT = COUNT - 1
1900 FOR M = 1 TO 2: SOUND = PEEK ( - 1633
6): NEXT M
1910 IF COUNT < 0 THEN MOVE$ = "OVER": VTB
23: HTAB 23: PRINT "I'll take ";N;" !"
;: FOR PAUSE = 1 TO 2000: NEXT PAUSE
1920 RETURN
1930 REM HUMAN'S MOVE
1940 POKE - 16368,0
1950 V = P - 128:N = V - 48
1960 IF (N = 1 OR N = 2 OR N = 3) AND N <
= T THEN MOVE$ = "OVER"
1970 IF MOVE$ = "" THEN PRINT CHR$ (7);
1980 RETURN
1990 REM FLAME 1
2000 FOR I = 1 TO 2
2010 P = PEEK ( - 16384)
2020 HPLLOT 22,47 TO 22,55: HPLLOT 23,46 TO
23,55
2030 NEXT
2040 RETURN
2050 REM FLAME 2
2060 FOR I = 1 TO 2
2070 P = PEEK ( - 16384)
2080 HPLLOT 23,34 TO 23,45: HPLLOT 24,37 TO
24,51
2090 NEXT
2100 RETURN
2110 REM TAKE STICKS AWAY
2120 HCOLOR= C1
2130 FOR I = T TO T - N + 1 STEP - 1
2140 GOSUB 2180
2150 NEXT I
2160 T = T - N
2170 RETURN
2180 REM ERASE
2190 X0 = 5 + I * 15 - 1
2200 X1 = 5 + I * 15 + 7
2210 FOR J = X0 TO X1

```

```
2220 HPlot J,59 TO J,140
2230 NEXT J
2240 RETURN
2250 REM END OF GAME
2260 TEXT : HOME
2270 VTab 1: HTAB 11: FLASH : PRINT "CONGR
      ATULATIONS !": NORMAL
2280 VTab 3: HTAB 1: PRINT "You picked up
      the flaming match,
2290 PRINT WINNER$;", and can therefore li
      ght
2300 PRINT "your victory cake.
2310 PRINT
2320 PRINT "So sorry, ";NM$(PLAYER);"."
2330 RETURN
```

Word Hunt

Original Game Concept by Robert W. Baker

This is a computer version of the newspaper and game-book puzzles where you find words in a mass of letters. Originally written for the Commodore PET, this version lets you enter your own word list. You're competing against the clock, as well as your opponent.

Just as in those puzzles you've seen in books and newspapers, "Word Hunt" lets you search for words hidden in a maze of letters. The computer does much of the work for you, creating the puzzle by inserting your word list among other letters chosen at random.

Type in and save Word Hunt. Run it, and you'll see instructions to set the CAPS LOCK key. On the Apple IIc, just press the key (found on the lower left of the keyboard). It acts as a toggle. If you press it again, the function is switched off. This option does not apply to the Apple II, II+, and IIe.

Word Hunt allows from one to nine players. Enter the number of players and each player's name. Next, select the difficulty level. The only difference from level to level is the time you're given to find a word. The easy level gives you 45 seconds, medium allows 30 seconds, and the hard level gives only 15 seconds to enter a word from the puzzle.

Type in the ten words you want to place in the puzzle. Each word must be at least three letters and no more than eight letters long. Depending on how hard you want to make the game, you may want to agree beforehand to a word-length limit. In fact, you don't even have to use words. Abbreviations and acronyms are fine. For an interesting twist, enter words from a foreign language. It'll make learning fun.

Puzzles

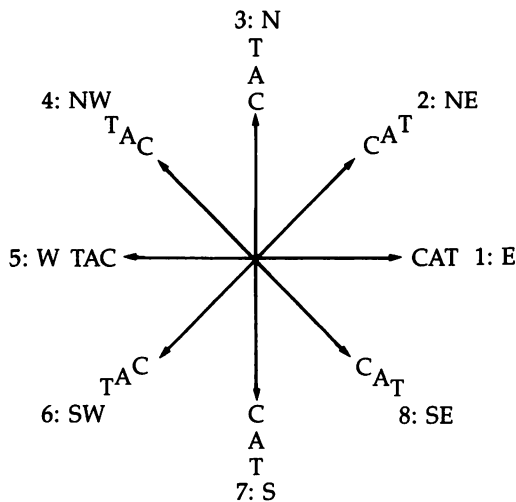
It will take the computer a few moments to create the puzzle. Be patient. If you entered MARINE, CORPS, AMERICA, MICRO, APPLE, WORD, HUNT, COMPUTE!, CAT, and DOG for your ten-word list, you would see a 10 × 10 letter block similar to Figure 1.

Figure 1. Word Hunt Puzzle

		Columns											
Rows		0	1	2	3	4	5	6	7	8	9		
	0	Y	A	F	A	C	I	R	E	M	A	1	E
	1	I	Z	G	W	R	E	W	C	F	W	2	NE
	2	X	Z	D	T	L	W	K	C	A	R	3	N
	3	O	J	Z	N	B	O	C	O	Y	T	4	NW
	4	R	M	T	U	A	R	O	M	E	X	5	W
	5	C	D	A	H	D	D	R	P	A	P	6	SW
	6	I	H	Y	R	Y	C	P	U	G	A	7	S
	7	M	X	F	L	I	F	S	T	O	R	8	SE
	8	F	T	P	T	F	N	A	E	D	A		
	9	A	P	P	L	E	J	E	!	H	E		

When you've found a word, enter the row and column co-ordinates of its first letter and the direction in which it runs. To key in the direction, hit the number corresponding to the appropriate compass heading, as Figure 2 shows. Words can appear in any direction. The figure shows how CAT would appear running in each direction.

Figure 2. Word Hunt Directions



If you were asked to find the word *COMPUTE!* in the puzzle example of Figure 1, you'd enter 2 for the row, 7 for the column, and then 7 for the direction, south. How many other words can you find in Figure 1?

When you locate a word, you score one point for each second left on the clock, and you lose ten points for each word you fail to find. To remind you that time is almost up, the computer starts to click when you have only five seconds remaining.

Since up to nine players are allowed in Word Hunt, you can take turns playing with the same list. The computer, of course, scrambles the words differently each time. The player with the most points at the end of the game wins.

The computer, by the way, checks for duplicate locations of a word and allows you to enter either one. Due to chance, this happens more often than you might imagine.

Word Hunt

```

100 REM WORD HUNT
110 REM INITIALIZE
120 GOSUB 180
130 REM PLAY GAME
140 GOSUB 600
150 REM DISPLAY RESULT
160 GOSUB 3690
170 END
180 REM INITIALIZE
190 REM TITLE
200 GOSUB 260
210 REM INSTRUCTIONS
220 GOSUB 310
230 REM KEY VALUES
240 GOSUB 450
250 RETURN
260 REM TITLE
270 PRINT CHR$(21): TEXT : HOME
280 VTAB 13: HTAB 15: PRINT "Word Hunt
290 FOR PAUSE = 1 TO 2000: NEXT
300 RETURN
310 REM INSTRUCTIONS
320 HOME
330 PRINT "Try to locate ten words as quic
    kly as
340 PRINT "possible in a maze of letters.
350 PRINT
360 PRINT "When you've found one, enter th
    e

```

```

370 PRINT "coordinates of its first letter
    , and
380 PRINT "the direction in which it lies.

390 PRINT
400 PRINT "Please set your "; INVERSE : PRINT
    "CAPS LOCK";: NORMAL : PRINT " key to"

410 PRINT "uppercase before continuing.
420 VTAB 23: HTAB 14: PRINT "Press any key
    ";
430 GET S$
440 RETURN
450 REM KEY VALUES
460 REM # OF WORDS
470 DATA 10
480 READ N
490 DIM W$(N),R$(N,N),P(N,3)
500 REM POINTS
510 FOR I = 1 TO 9
520 PT(I) = 0
530 NEXT
540 REM DIRECTIONS
550 DATA E,NE,N,NW,W,SW,S,SE
560 FOR I = 1 TO 8
570 READ DR$(I)
580 NEXT
590 RETURN
600 REM PLAY GAME
610 REM PLAYERS
620 GOSUB 740
630 REM CHOOSE SKILL LEVEL
640 GOSUB 920
650 REM ENTER WORDS
660 GOSUB 1060
670 REM SORT THEM
680 GOSUB 1180
690 REM PLAY
700 FOR Q = 1 TO NP
710 GOSUB 1250
720 NEXT Q
730 RETURN
740 REM PLAYERS
750 HOME
760 VTAB 1: HTAB 1: PRINT "How many player
    s are there (1 to 9) ? "; CHR$(7);
770 GET S$
780 NP = VAL (S$)
790 IF NP < 1 OR NP > 9 THEN 760
800 N$ = "names"

```



```

810 IF NP = 1 THEN N$ = "name"
820 VTAB 3: HTAB 1: PRINT "Please enter yo
ur ";N$;". Up to 15
830 PRINT "characters are allowed.
840 FOR I = 1 TO NP
850 VTAB 6: HTAB 18: PRINT SPC( 25); CHR$
(7)
860 VTAB 6: HTAB 1: PRINT "Player No. ";I;

870 INPUT " = ? ";N$
880 IF N$ = "" OR LEN (N$) > 15 THEN 850
890 N$(I) = N$
900 NEXT
910 RETURN
920 REM SKILL LEVEL
930 HOME
940 VTAB 7: HTAB 5: PRINT "Seconds Allowed
to Find a Word
950 INVERSE
960 FOR I = 1 TO 3: VTAB 2 * I + 7: HTAB 1
0: PRINT I: NEXT
970 NORMAL
980 VTAB 9: HTAB 12: PRINT "Easy"; TAB( 25
)"45
990 VTAB 11: HTAB 12: PRINT "Medium"; TAB(
25)"30
1000 VTAB 13: HTAB 12: PRINT "Hard"; TAB(
25)"15
1010 VTAB 16: HTAB 8: PRINT "Choice = ? ";
CHR$ (7);
1020 GET S$
1030 SK = VAL (S$)
1040 IF SK < 1 OR SK > 3 THEN 1010
1050 RETURN
1060 REM ENTER WORDS
1070 HOME
1080 PRINT "Please enter your words (3 to
8 letters"
1090 PRINT "long). "
1100 FOR I = 1 TO N
1110 VTAB 4: HTAB 16: PRINT SPC( 25); CHR$
(7)
1120 VTAB 4: HTAB 1: PRINT "Word No. ";I; TAB(
12)"= ";
1130 INPUT "? ";W$
1140 IF LEN (W$) < 3 OR LEN (W$) > 8 THEN
1110
1150 W$(I) = W$
1160 NEXT
1170 RETURN

```

```

1180 REM SORT THEM
1190 SWITCH$ = "N"
1200 FOR J = 1 TO N - 1
1210 IF LEN (W$(J)) < LEN (W$(J + 1)) THEN
    H$ = W$(J):W$(J) = W$(J + 1):W$(J + 1)
    = H$:SWITCH$ = "Y"
1220 NEXT
1230 IF SWITCH$ = "Y" THEN 1190
1240 RETURN
1250 REM PLAY
1260 REM CREATE PUZZLE
1270 GOSUB 1420
1280 REM FILL IN BLANKS
1290 GOSUB 2470
1300 REM CALL PLAYER
1310 HOME
1320 PRINT "Your turn ";N$(Q);"."
1330 VTAB 3: HTAB 14: INVERSE : PRINT "PRE
    SS ANY KEY";: NORMAL : PRINT " "; CHR$
    (7);

1340 GET S$
1350 REM DRAW BOARD
1360 GOSUB 2530
1370 REM GUESS WORDS
1380 FOR I = 1 TO N
1390 GOSUB 2880
1400 NEXT I
1410 RETURN
1420 REM CREATE PUZZLE
1430 REM KEY VALUES
1440 GOSUB 1610
1450 REM PLACE WORDS
1460 FOR I = 1 TO N
1470 GOSUB 1740
1480 IF SR$ = "" THEN I = N
1490 NEXT I
1500 REM ASK TO TRY AGAIN
1510 IF SR$ = "" THEN GOSUB 1540
1520 IF S$ = "Y" THEN 1440
1530 RETURN
1540 REM ASK TO TRY AGAIN
1550 VTAB 22: HTAB 1: PRINT "I can't fit y
    our words. Should I try
1560 VTAB 23: HTAB 1: PRINT "again (Y/N) ?
    "; CHR$ (7);
1570 GET S$
1580 IF S$ = "N" THEN STOP
1590 IF S$ < > "Y" THEN 1560
1600 RETURN
1610 REM KEY VALUES

```

```

1620 HOME
1630 VTAB 12: HTAB 13: FLASH : PRINT "CREA
      TING PUZZLE": NORMAL
1640 FOR I = 0 TO N - 1
1650 FOR J = 0 TO N - 1
1660 R$(I,J) = "*"
1670 NEXT J,I
1680 REM COORDINATES OF FIRST LETTER, & D
      IRECTION
1690 FOR I = 1 TO N
1700 FOR J = 1 TO 3
1710 P(I,J) = 0
1720 NEXT J,I
1730 RETURN
1740 REM PLACE WORDS
1750 DEF FN R(V) = INT (10 * RND (V))
1760 LN = LEN (W$(I)):SR$ = ""
1770 FOR CNT = 1 TO 100
1780 REM POSITION OF FIRST LETTER
1790 R = FN R(1):C = FN R(1)
1800 REM TRY DIRECTIONS
1810 FOR L = 1 TO 8
1820 D = INT (8 * RND (1)) + 1
1830 REM TRY TO INSERT WORD
1840 GOSUB 1880
1850 IF SR$ = "OKAY" THEN L = 8: CNT = 100
1860 NEXT L, CNT
1870 RETURN
1880 REM INSERT WORD
1890 REM CHECK LENGTH
1900 GOSUB 2010
1910 IF G$ = "" THEN 2000
1920 REM GET CHARACTERS
1930 GOSUB 2120
1940 REM EXAMINE THEM
1950 GOSUB 2260
1960 IF P$ = "TAKEN" THEN 2000
1970 REM INSERT
1980 GOSUB 2330
1990 SR$ = "OKAY"
2000 RETURN
2010 REM LENGTH
2020 G$ = ""
2030 IF D = 1 AND LN < = 10 - C THEN G$ =
      "OKAY"
2040 IF D = 2 AND LN < = 10 - C AND LN <
      = R + 1 THEN G$ = "OKAY"
2050 IF D = 3 AND LN < = R + 1 THEN G$ =
      "OKAY"
2060 IF D = 4 AND LN < = R + 1 AND LN < =
      C + 1 THEN G$ = "OKAY"

```

```

2070 IF D = 5 AND LN < = C + 1 THEN G$ =
"OKAY"
2080 IF D = 6 AND LN < = C + 1 AND LN < =
10 - R THEN G$ = "OKAY"
2090 IF D = 7 AND LN < = 10 - R THEN G$ =
"OKAY"
2100 IF D = 8 AND LN < = 10 - C AND LN <
= 10 - R THEN G$ = "OKAY"
2110 RETURN
2120 REM CHARACTERS
2130 S$ = ""
2140 FOR J = 1 TO LN
2150 IF D = 1 THEN A$ = R$(R, C + J - 1)
2160 IF D = 2 THEN A$ = R$(R - J + 1, C + J
- 1)
2170 IF D = 3 THEN A$ = R$(R - J + 1, C)
2180 IF D = 4 THEN A$ = R$(R - J + 1, C - J
+ 1)
2190 IF D = 5 THEN A$ = R$(R, C - J + 1)
2200 IF D = 6 THEN A$ = R$(R + J - 1, C - J
+ 1)
2210 IF D = 7 THEN A$ = R$(R + J - 1, C)
2220 IF D = 8 THEN A$ = R$(R + J - 1, C + J
- 1)
2230 S$ = S$ + A$
2240 NEXT J
2250 RETURN
2260 REM EXAMINE
2270 P$ = "OKAY"
2280 FOR J = 1 TO LN
2290 L$ = MID$(S$, J, 1)
2300 IF L$ < > "*" AND L$ < > MID$(W$(
I), J, 1) THEN P$ = "TAKEN"
2310 NEXT J
2320 RETURN
2330 REM INSERT
2340 FOR J = 1 TO LN
2350 L$ = MID$(W$(I), J, 1)
2360 IF D = 1 THEN R$(R, C + J - 1) = L$
2370 IF D = 2 THEN R$(R - J + 1, C + J - 1)
= L$
2380 IF D = 3 THEN R$(R - J + 1, C) = L$
2390 IF D = 4 THEN R$(R - J + 1, C - J + 1)
= L$
2400 IF D = 5 THEN R$(R, C - J + 1) = L$
2410 IF D = 6 THEN R$(R + J - 1, C - J + 1)
= L$
2420 IF D = 7 THEN R$(R + J - 1, C) = L$
2430 IF D = 8 THEN R$(R + J - 1, C + J - 1)
= L$

```

```

2440 NEXT J
2450 P(I,1) = R:P(I,2) = C:P(I,3) = D
2460 RETURN
2470 REM FILL IN BLANKS
2480 FOR I = 0 TO N - 1
2490 FOR J = 0 TO N - 1
2500 IF R$(I,J) = "*" THEN R$(I,J) = CHR$
(64 + INT (26 * RND (1) + 1))
2510 NEXT J,I
2520 RETURN
2530 REM DRAW BOARD
2540 REM HEADINGS
2550 GOSUB 2590
2560 REM BODY
2570 GOSUB 2770
2580 RETURN
2590 REM HEADINGS
2600 HOME
2610 REM DIRECTIONS
2620 FOR L = 1 TO 8
2630 VTAB L * 2 + 2: HTAB 36: PRINT L;" ";
: INVERSE : PRINT DR$(L): NORMAL
2640 NEXT L
2650 REM ROWS
2660 VTAB 9: HTAB 1: INVERSE : PRINT "R"
2670 PRINT "O": PRINT "W": PRINT "S"
2680 FOR L = 0 TO 9
2690 VTAB L * 2 + 3: HTAB 3: PRINT L
2700 NEXT L
2710 REM COLUMNS
2720 FOR L = 0 TO 9
2730 VTAB 1: HTAB L * 3 + 5: PRINT L
2740 NEXT L
2750 NORMAL
2760 RETURN
2770 REM BODY
2780 R = 3
2790 FOR I = 0 TO N - 1
2800 C = 5
2810 FOR J = 0 TO N - 1
2820 VTAB R: HTAB C: PRINT R$(I,J)
2830 C = C + 3
2840 NEXT J
2850 R = R + 2
2860 NEXT I
2870 RETURN
2880 REM GUESS
2890 REM VALUES
2900 GOSUB 3040
2910 REM ROW

```

```

2920 HT = 15: GOSUB 3200:R = V
2930 REM COLUMN
2940 IF TICS > 0 THEN HT = 22: GOSUB 3200:
    C = V
2950 REM DIRECTION
2960 IF TICS > 0 THEN HT = 29: GOSUB 3200:
    D = V
2970 REM CHECK ANSWER
2980 GOSUB 3410
2990 REM DISPLAY RESULT
3000 GOSUB 3530
3010 IF AN$ = "WRONG !" THEN GOSUB 3580
3020 IF AN$ = "RIGHT !" THEN GOSUB 3660
3030 RETURN
3040 REM VALUES
3050 VTAB 23: HTAB 1: PRINT SPC( 40)
3060 VTAB 23: HTAB 7: PRINT "Press "; INVERSE
    : PRINT "RETURN";: NORMAL : PRINT " fo
    r Word# ";I;" "; CHR$( 7);
3070 GET S$
3080 IF ASC (S$) < > 13 THEN 3060
3090 VTAB 23: HTAB 1: PRINT SPC( 40)
3100 INVERSE
3110 VTAB 23: HTAB 11: PRINT "ROW";: HTAB
    18: PRINT "COL";: HTAB 25: PRINT "DIR"
    ;: HTAB 32: PRINT "TIME"
3120 NORMAL
3130 REM SKILL LEVEL
3140 IF SK = 1 THEN TICS = 45
3150 IF SK = 2 THEN TICS = 30
3160 IF SK = 3 THEN TICS = 15
3170 R = - 9:C = - 9:D = - 9
3180 VTAB 23: HTAB 1: PRINT W$(I)
3190 RETURN
3200 REM GET RESPONSE
3210 M$ = ""
3220 IF TICS < 0 THEN TICS = 0
3230 VTAB 23: HTAB 37: PRINT SPC( 2);: HTAB
    37: PRINT INT (TICS)
3240 VTAB 23: HTAB HT: FLASH : PRINT "?";:
    NORMAL : HTAB HT
3250 FOR L = 1 TO 4
3260 P = PEEK ( - 16384)
3270 NEXT L
3280 TICS = TICS - 0.12
3290 IF TICS < = 5 THEN FOR L = 1 TO 2:S
    OUND = PEEK ( - 16336): NEXT L
3300 IF P > = 128 THEN GOSUB 3330
3310 IF TICS > 0.25 AND M$ < > "GOOD" THEN
    3220

```

```

3320 RETURN
3330 REM CHECK RESPONSE
3340 A = P - 128:V = A - 48
3350 IF (HT = 15 OR HT = 22) AND (V > = 0
    AND V < = 9) THEN M$ = "GOOD"
3360 IF (HT = 29) AND (V > 0 AND V < 9) THEN
    M$ = "GOOD"
3370 IF M$ = "GOOD" THEN PRINT V
3380 IF M$ = "" THEN PRINT CHR$ (7);
3390 POKE - 16368,0
3400 RETURN
3410 REM CHECK ANSWER
3420 AN$ = "WRONG !"
3430 IF P(I,1) = R AND P(I,2) = C AND P(I,
    3) = D THEN AN$ = "RIGHT !"
3440 IF AN$ = "WRONG !" THEN GOSUB 3460
3450 RETURN
3460 REM CHECK FOR DUPLICATE ANSWER
3470 REM LENGTH
3480 LN = LEN (W$(I)): GOSUB 2010
3490 REM WORD
3500 IF G$ = "OKAY" THEN GOSUB 2120
3510 IF G$ = "OKAY" AND S$ = W$(I) THEN AN
    $ = "RIGHT !"
3520 RETURN
3530 REM RESULT
3540 VTAB 23: HTAB 1: PRINT SPC( 40)
3550 VTAB 23: HTAB 17: FLASH : PRINT AN$;:
    NORMAL
3560 FOR PAUSE = 1 TO 20: PRINT CHR$ (7);
    : NEXT PAUSE
3570 RETURN
3580 REM WRONG
3590 R = 2 * P(I,1) + 3:C = 3 * P(I,2) + 5
3600 VTAB R: HTAB C: FLASH : PRINT R$(P(I,
    1),P(I,2))
3610 FOR PAUSE = 1 TO 3000: NEXT PAUSE
3620 NORMAL
3630 VTAB R: HTAB C: PRINT R$(P(I,1),P(I,2
    ))
3640 PT(Q) = PT(Q) - 10
3650 RETURN
3660 REM RIGHT
3670 PT(Q) = PT(Q) + TICS
3680 RETURN
3690 REM FINAL SCORE
3700 HOME
3710 PRINT TAB( 15)"FINAL SCORE"
3720 INVERSE : VTAB 3: HTAB 5: PRINT "PLAY
    ER";: HTAB 25: PRINT "TOTAL POINTS": NORMAL

```

```
3730 FOR I = 1 TO NP
3740 S$ = STR$ ( INT (PT(I)))
3750 VTAB I + 4: HTAB 5: PRINT N$(I); TAB(
      32 - LEN (S$))S$
3760 NEXT
3770 VTAB 23: HTAB 14: PRINT "Press any ke
      y "; CHR$ (7);
3780 GET S$
3790 RETURN
```


Solitaire Checkers

In this solitaire version of checkers, you try to remove as many of the pieces as possible. It may be simple to play, but it takes skill and planning to master.

"Solitaire Checkers," a challenging contest of logic, places 40 pieces around the edges of a 7×7 board. You try to eliminate as many as possible by leaping over the checkers diagonally, just as in the traditional board game.

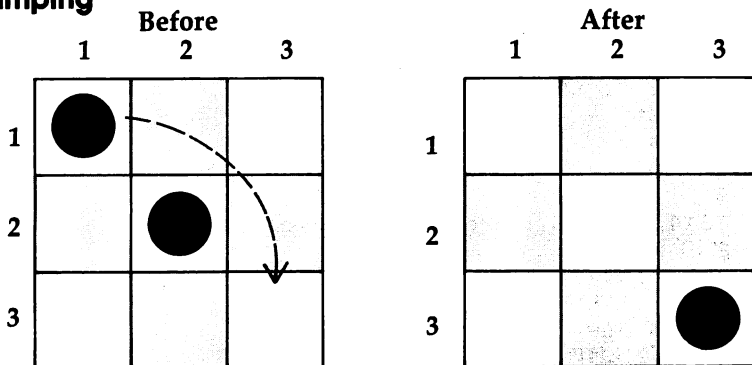
Removing all but 11 or 12 checkers is relatively easy. Ending up with only a handful requires the insight of a mathematician and the foresight of a soothsayer.

The Checkerboard Screen

After typing in and saving a copy of Solitaire Checkers, run it. A short instruction screen appears and asks what kind of monitor you have; answer Y if you have a color monitor, and N if not. Next, the board is drawn. Notice the row numbers on the left and the column numbers above the board. You'll use these numbers to select and move your pieces.

Making a move is simple. First, type in the row and column coordinates of the checker you want to move. A new prompt will appear, asking for the row and column numbers of the square to which you want to move that checker. The figure illustrates a sample move.

Jumping



When checker 1,1 is moved to square 3,3 positions 1,1 and 2,2 become vacant. The jumped checker 2,2 has been removed from play.

The piece at row 1, column 1 is moved to row 3, column 3. The piece at 2,2 is then removed from play. Now both positions 1,1 and 2,2 are vacant. You must *always* jump a piece.

If you decide to jump a checker and then change your mind, don't worry. Simply enter an illegal position as the destination square. For instance, type in the coordinates of a vacant square on the other side of the board. You'll see the message *ILLEGAL MOVE!* and a buzzer will sound for a few seconds, but that's all. You can then make your desired move.

The computer checks the board after each move to see if further play is possible. When it isn't, you'll see how many pieces are left, as well as the computer's ranking of your performance.

Checkers Left	Rank
1-5	Master
6-10	Apprentice
11 or more	Novice

Technical note. If you have an Apple II or II+ with little memory, you may encounter an OUT OF MEMORY error when you run this program. If so, eliminate lines 100, 430-500, 1050-1090, and 3330-3430. This will solve the problem.

Solitaire Checkers

```

100 LOMEM: 16384
110 REM SOLITAIRE CHECKERS
120 REM INITIALIZE
130 GOSUB 230
140 REM PLAY GAME
150 GOSUB 570
160 REM PLAY AGAIN
170 VTAB 24: HTAB 12: PRINT "Play again (Y
    /N) ? "; CHR$(7);
180 GET S$
190 IF S$ = "Y" OR S$ = "y" THEN 150
200 IF S$ < > "N" AND S$ < > "n" THEN 17
    0
210 TEXT : HOME : PRINT "Bye-bye."
220 END
230 REM INITIALIZE
240 REM TITLE
250 GOSUB 360
260 REM INSTRUCTIONS
270 GOSUB 410
280 K = 15
290 DIM B(7,7),C1(K),C2(K)

```

```

300 REM CHECKER POINTS
310 DATA 5,9,3,11,2,12,1,13,1,13,0,14,0,1
    4,0,14,0,14,0,14,1,13,1,13,2,12,3,11,5
    ,9
320 FOR I = 1 TO K
330 READ C1(I),C2(I)
340 NEXT
350 RETURN
360 REM TITLE
370 PRINT CHR$(21): TEXT : HOME
380 VTAB 12: HTAB 11: PRINT "Solitaire Che
    ckers
390 FOR PAUSE = 1 TO 2000: NEXT
400 RETURN
410 REM INSTRUCTIONS
420 HOME
430 PRINT "I'm going to place 40 checkers
    around
440 PRINT "the outer two borders of a 7x7
    board.
450 PRINT
460 PRINT "Try to remove as many pieces as
    you can.";
470 PRINT
480 PRINT "A checker is lifted from play w
    hen it's"
490 PRINT "jumped diagonally, just like in
    the
500 PRINT "regular game.
510 VTAB 11: HTAB 1: PRINT "Are you using
    a color monitor (Y/N) ? "; CHR$(7);
520 GET S$
530 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 510
540 K1 = 0:K2 = 0:K3 = 3:K4 = 3
550 IF S$ = "Y" OR S$ = "y" THEN K1 = 3:K2
    = 3:K3 = 5:K4 = 6
560 RETURN
570 REM PLAY GAME
580 REM INITIAL POSITION
590 GOSUB 700
600 REM ENTER MOVE
610 GOSUB 1520
620 REM MOVE PIECES
630 GOSUB 2220
640 REM CHECK FOR END
650 GOSUB 2340
660 IF GAME$ = "ON" THEN 610
670 REM END
680 GOSUB 2940

```

```

690 RETURN
700 REM INITIALIZE
710 REM RECORD INITIAL POSITION
720 GOSUB 820
730 REM LABEL BOARD
740 GOSUB 910
750 REM DRAW BOXES
760 GOSUB 1110
770 REM DRAW LINES
780 GOSUB 1270
790 REM DRAW CHECKERS
800 GOSUB 1380
810 RETURN
820 REM RECORD
830 REM 1=FILLED; 0=VACANT
840 FOR R = 1 TO 7
850 FOR C = 1 TO 7
860 B(R,C) = 1
870 IF (R > 2 AND R < 6) AND (C > 2 AND C <
    6) THEN B(R,C) = 0
880 NEXT C,R
890 CHECKERS = 40
900 RETURN
910 REM LABEL
920 HOME
930 HGR : HCOLOR= 0: HPLLOT 1,1: CALL 62454

940 REM COLUMNS
950 HCOLOR= 3
960 FOR I = 1 TO 7
970 X = 30 * I + 20:Y = 10
980 ON I GOSUB 3040,3070,3120,3160,3200,32
    40,3290
990 NEXT
1000 REM ROWS
1010 FOR I = 1 TO 7
1020 X = 24:Y = 20 * I + 10
1030 ON I GOSUB 3040,3070,3120,3160,3200,3
    240,3290
1040 NEXT
1050 REM LETTERS
1060 FOR I = 1 TO 3
1070 X = 7:Y = 10 * I + 65
1080 ON I GOSUB 3330,3370,3400
1090 NEXT
1100 RETURN
1110 REM DRAW BOXES
1120 FOR R = 1 TO 7
1130 FOR C = 1 TO 7
1140 GOSUB 1170: GOSUB 1210

```

```

1150 NEXT C,R
1160 RETURN
1170 REM COLOR OF BOX
1180 HCOLOR= K4
1190 IF (R + C) / 2 = INT ((R + C) / 2) THEN
    HCOLOR= K3
1200 RETURN
1210 REM BOX
1220 X = C * 30 + 8:Y = R * 20 - 2
1230 FOR I = X TO X + 28
1240 HPLLOT I,Y TO I,Y + 18
1250 NEXT I
1260 RETURN
1270 REM DRAW LINES
1280 REM HORIZONTAL
1290 HCOLOR= K2
1300 FOR I = 17 TO 157 STEP 20
1310 HPLLOT 37,I TO 247,I
1320 NEXT
1330 REM VERTICAL
1340 FOR I = 37 TO 247 STEP 30
1350 HPLLOT I,17 TO I,157
1360 NEXT
1370 RETURN
1380 REM DRAW CHECKERS
1390 HCOLOR= K1
1400 FOR R = 1 TO 7
1410 FOR C = 1 TO 7
1420 IF B(R,C) = 1 THEN GOSUB 1450
1430 NEXT C,R
1440 RETURN
1450 REM CHECKERS
1460 X = C * 30 + 15:Y = R * 20
1470 FOR I = 1 TO K
1480 HPLLOT X + C1(I),Y TO X + C2(I),Y
1490 Y = Y + 1
1500 NEXT
1510 RETURN
1520 REM ENTER MOVE
1530 REM CHECKER TO MOVE
1540 GOSUB 1630
1550 REM MAKE IT SQUARE
1560 GOSUB 1860
1570 REM PLACE TO PUT IT
1580 GOSUB 1980
1590 REM CHECK FOR LEGAL MOVE
1600 GOSUB 2060
1610 IF R# = "ILLEGAL" THEN 1540
1620 RETURN
1630 REM CHECKER TO MOVE

```

```

1640 A$ = "Piece to move: "
1650 GOSUB 1710
1660 REM MAKE SURE PIECE IS THERE
1670 VTAB 23: HTAB 18: PRINT SPC( 8)
1680 IF B(R,C) = 0 THEN VTAB 23: HTAB 18:
    INVERSE : PRINT "VACANT !": NORMAL : GOTO
    1650
1690 R1 = R:C1 = C
1700 RETURN
1710 REM CHOOSE ROW & COLUMN
1720 VTAB 21: HTAB 1: PRINT SPC( 40)
1730 REM ROW
1740 VTAB 21: HTAB 1: PRINT A$;"Row = ? ";
    CHR$( 7);
1750 GET S$
1760 R = VAL (S$)
1770 IF R < 1 OR R > 7 THEN 1740
1780 PRINT R
1790 REM COLUMN
1800 VTAB 21: HTAB 27: PRINT "Column = ? "
    ; CHR$( 7);
1810 GET S$
1820 C = VAL (S$)
1830 IF C < 1 OR C > 7 THEN 1800
1840 PRINT C
1850 RETURN
1860 REM MAKE IT SQUARE
1870 REM ERASE CHECKER
1880 GOSUB 1170: GOSUB 1450
1890 REM DRAW SQUARE
1900 HCOLOR= K1: GOSUB 1920
1910 RETURN
1920 REM SQUARE
1930 X = C * 30 + 18:Y = R * 20 + 3
1940 FOR I = 0 TO 7
1950 HPLOT X + I,Y TO X + I,Y + 8
1960 NEXT
1970 RETURN
1980 REM PLACE TO PUT IT
1990 A$ = "Place to put: "
2000 GOSUB 1710
2010 REM MAKE SURE SPACE IS VACANT
2020 VTAB 23: HTAB 18: PRINT SPC( 8)
2030 IF B(R,C) < > 0 THEN VTAB 23: HTAB
    18: INVERSE : PRINT "FILLED !": NORMAL
    : GOTO 2000
2040 R2 = R:C2 = C
2050 RETURN
2060 REM LEGALITY
2070 R$ = "":RM = R2:CM = C2

```

```

2080 IF R1 > R2 THEN RM = R1
2090 RM = RM - 1
2100 IF C1 > C2 THEN CM = C1
2110 CM = CM - 1
2120 IF ABS (R2 - R1) = 2 AND ABS (C2 -
    C1) = 2 AND B(RM,CM) = 1 THEN 2140
2130 R$ = "ILLEGAL": GOSUB 2150
2140 RETURN
2150 REM ILLEGAL MOVE
2160 VTAB 23: HTAB 15: FLASH : PRINT "ILLE
    GAL MOVE !": NORMAL
2170 FOR PAUSE = 1 TO 500: SOUND = PEEK ( -
    16336): NEXT
2180 REM REDRAW CHECKER
2190 R = R1: C = C1: GOSUB 1450
2200 VTAB 23: HTAB 15: PRINT SPC( 14)
2210 RETURN
2220 REM MOVE
2230 REM STRIKE 2 PIECES
2240 R = R1: C = C1: GOSUB 1170: GOSUB 1450
2250 R = RM: C = CM: GOSUB 1450
2260 REM ADD PIECE
2270 R = R2: C = C2: HCOLOR= K1: GOSUB 1450
2280 REM RECORD
2290 B(R1,C1) = 0
2300 B(RM,CM) = 0
2310 B(R2,C2) = 1
2320 CHECKERS = CHECKERS - 1
2330 RETURN
2340 REM CHECK FOR END
2350 GAME$ = ""
2360 REM NEGATIVE SLANT
2370 GOSUB 2430
2380 IF GAME$ = "" THEN GOSUB 2550
2390 REM POSITIVE (/) SLANT
2400 IF GAME$ = "" THEN GOSUB 2670
2410 IF GAME$ = "" THEN GOSUB 2790
2420 RETURN
2430 REM -; LOWER TRIANGLE
2440 E = 1
2450 FOR I = 5 TO 1 STEP - 1
2460 FOR J = 1 TO E
2470 X = B(I + J - 1,J)
2480 Y = B(I + J,J + 1)
2490 Z = B(I + J + 1,J + 2)
2500 GOSUB 2910
2510 NEXT J
2520 E = E + 1
2530 NEXT I
2540 RETURN

```

```

2550 REM -; UPPER TRIANGLE
2560 E = 1
2570 FOR I = 5 TO 2 STEP - 1
2580 FOR J = 1 TO E
2590 X = B(J,I + J - 1)
2600 Y = B(J + 1,I + J)
2610 Z = B(J + 2,I + J + 1)
2620 GOSUB 2910
2630 NEXT J
2640 E = E + 1
2650 NEXT I
2660 RETURN
2670 REM +; UPPER TRIANGLE
2680 E = 1
2690 FOR I = 3 TO 7
2700 FOR J = 1 TO E
2710 X = B(I - J + 1,J)
2720 Y = B(I - J,J + 1)
2730 Z = B(I - J - 1,J + 2)
2740 GOSUB 2910
2750 NEXT J
2760 E = E + 1
2770 NEXT I
2780 RETURN
2790 REM +; LOWER TRIANGLE
2800 E = 1
2810 FOR I = 5 TO 2 STEP - 1
2820 FOR J = 1 TO E
2830 X = B(8 - J,I + J - 1)
2840 Y = B(8 - J - 1,I + J)
2850 Z = B(8 - J - 2,I + J + 1)
2860 GOSUB 2910
2870 NEXT J
2880 E = E + 1
2890 NEXT I
2900 RETURN
2910 REM CHECK
2920 IF (X = 1 AND Y = 1 AND Z = 0) OR (Z =
    1 AND Y = 1 AND X = 0) THEN GAME$ = "O
    N"
2930 RETURN
2940 REM OVER
2950 VTAB 21: HTAB 1: PRINT SPC( 40)
2960 VTAB 23: HTAB 13: FLASH : PRINT "THE
    GAME'S OVER !": NORMAL
2970 FOR PAUSE = 1 TO 30: PRINT CHR$( 7);
    : NEXT
2980 IF CH < = 5 THEN RK$ = "Master!"
2990 IF CH > 5 AND CH < = 10 THEN RK$ = "
    Apprentice."

```



```

3000 IF CH > 10 THEN RK$ = "Novice."
3010 VTAB 23: HTAB 13: PRINT SPC( 17)
3020 VTAB 21: HTAB 6: PRINT CH;" checkers
      remain. ";RK$
3030 RETURN
3040 REM 1
3050 HPLLOT X + 1,Y TO X + 3,Y: HPLLOT X + 2
      ,Y TO X + 2,Y - 6: HPLLOT X + 1,Y - 5
3060 RETURN
3070 REM 2
3080 HPLLOT X,Y TO X + 4,Y: HPLLOT X,Y - 1: HPLLOT
      X + 1,Y - 2
3090 HPLLOT X + 2,Y - 3 TO X + 3,Y - 3: HPLLOT
      X + 4,Y - 4 TO X + 4,Y - 5
3100 HPLLOT X + 1,Y - 6 TO X + 3,Y - 6: HPLLOT
      X,Y - 5
3110 RETURN
3120 REM 3
3130 HPLLOT X,Y - 6 TO X + 4,Y - 6: HPLLOT X
      + 1,Y TO X + 3,Y: HPLLOT X + 2,Y - 3 TO
      X + 3,Y - 3
3140 HPLLOT X,Y - 1: HPLLOT X + 4,Y - 1 TO X
      + 4,Y - 2: HPLLOT X + 3,Y - 4: HPLLOT X
      + 4,Y - 5
3150 RETURN
3160 REM 4
3170 HPLLOT X + 3,Y TO X + 3,Y - 6: HPLLOT X
      ,Y - 2 TO X + 4,Y - 2
3180 HPLLOT X,Y - 3: HPLLOT X + 1,Y - 4: HPLLOT
      X + 2,Y - 5
3190 RETURN
3200 REM 5
3210 HPLLOT X,Y - 6 TO X + 4,Y - 6: HPLLOT X
      ,Y - 4 TO X + 3,Y - 4: HPLLOT X + 4,Y -
      1 TO X + 4,Y - 3: HPLLOT X + 1,Y TO X +
      3,Y
3220 HPLLOT X,Y - 5: HPLLOT X,Y - 1
3230 RETURN
3240 REM 6
3250 HPLLOT X + 1,Y TO X + 3,Y: HPLLOT X,Y -
      3 TO X + 3,Y - 3: HPLLOT X + 2,Y - 6 TO
      X + 4,Y - 6
3260 HPLLOT X + 1,Y - 5: HPLLOT X,Y - 4
3270 HPLLOT X,Y - 1 TO X,Y - 2: HPLLOT X + 4
      ,Y - 1 TO X + 4,Y - 2
3280 RETURN
3290 REM 7
3300 HPLLOT X + 1,Y TO X + 1,Y - 2: HPLLOT X
      ,Y - 6 TO X + 4,Y - 6

```

```

3310 HPLOT X + 2,Y - 3: HPLOT X + 3,Y - 4:
      HPLOT X + 4,Y - 5
3320 RETURN
3330 REM R
3340 HPLOT X,Y TO X,Y - 6: HPLOT X + 1,Y -
      6 TO X + 3,Y - 6: HPLOT X + 1,Y - 3 TO
      X + 3,Y - 3: HPLOT X + 4,Y - 4 TO X +
      4,Y - 5
3350 HPLOT X + 2,Y - 2: HPLOT X + 3,Y - 1:
      HPLOT X + 4,Y
3360 RETURN
3370 REM O
3380 HPLOT X,Y - 1 TO X,Y - 5: HPLOT X + 4
      ,Y - 1 TO X + 4,Y - 5: HPLOT X + 1,Y TO
      X + 3,Y: HPLOT X + 1,Y - 6 TO X + 3,Y -
      6
3390 RETURN
3400 REM W
3410 HPLOT X,Y TO X,Y - 6: HPLOT X + 4,Y TO
      X + 4,Y - 6
3420 HPLOT X + 1,Y - 1: HPLOT X + 3,Y - 1:
      HPLOT X + 2,Y - 2: HPLOT X + 2,Y - 3
3430 RETURN

```

Mosaic Puzzle

Original Game Concept by Bruce Jordan

This program is an electronic version of the sliding puzzle. The object is to slide the numerals all around to get them into the desired order. The idea is simple—but sometimes maddening.

Long before Rubik's Cube, sliding-square puzzles challenged the mind. "Mosaic Puzzle" is an electronic version of those puzzles you may have spent hours trying to figure out. Unlike the plastic version, this computer game keeps track of the time and the number of moves you make. You can even save an uncompleted board to disk and return to it later.

You're trying to rearrange the numbers of a 3×3 square, using as few moves and as little time as possible. The square consists of eight numbers and a blank pane, with the computer scrambling these pieces with each new game. Your job is to put them back in order.

Move and Slide

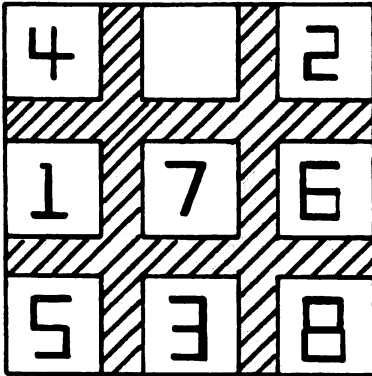
Type in and save Mosaic Puzzle. Run the program and a short display appears, asking for a random number seed. Type in any number from 1 to 1000. This insures that the starting board will be different from session to session. (If you want to compete against another player, you could both enter the same number seed, then see who can complete the puzzle the fastest, with the fewest moves.)

You'll be asked if you're using a color monitor and if you are resuming an old game. Enter Y or N for each question. Of course, you must have previously saved a game to disk in order to resume it. If this is your first game, type N. The board is then drawn (it takes a few moments, so be patient) and the clock starts. Take a look at the figure for a moment. It shows a board you might begin with, as well as the board you want to create.

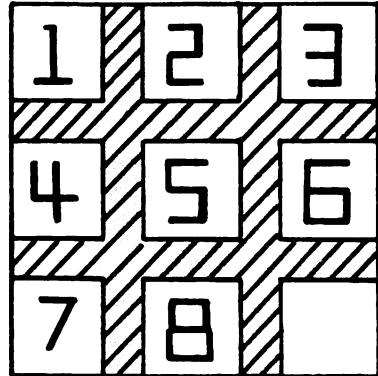
You move pieces in Mosaic Puzzle by sliding them to and fro, as if they were locked onto a flat playing board. Lifting or jumping pieces is impossible. It makes your job a little more difficult, doesn't it? Four keys slide the pieces left, right, up, or down. The keys and their directions are shown at the bottom of the screen.

Boards

Your Board



Your Goal



D Left F Right L Up M Down Q = Quit

There's always just one pane you can move in any individual direction. In the figure, for example, there are three panes that can move, but only one that can move right. That's the 4 pane. Pressing the F key moves the 4 pane to the right. The blank pane is then in the upper left of the board. If you'd pressed D instead, the 2 pane would have moved left.

Reaching your goal isn't easy. Unless you constantly think ahead, you'll endlessly rearrange pieces you thought were already set. If you despair of ever finding a solution, give yourself a break by pressing Q for Quit. Save your board to the disk. All you have to do is provide a filename for the game you saved (such as GAME1, GAME2, and so on). You can return to it at any time by running the program and entering Y at the *resuming old game* prompt.

The computer keeps track of the number of moves it takes you to finish. Try playing Mosaic Puzzle with your friends. The player using the fewest moves is the winner.

Mosaic Puzzle

```

100 REM MOSAIC PUZZLE
110 REM INITIALIZE
120 GOSUB 240
130 REM PLAY GAME
140 GOSUB 750
150 REM RESULT
160 IF GAME$ = "QUIT" THEN GOSUB 2810
170 IF GAME$ = "OVER" THEN GOSUB 3040
    
```

```

180 VTAB 24: HTAB 12: PRINT "Play again (Y
/N) ? "; CHR$ (7);
190 GET S$
200 IF S$ = "Y" OR S$ = "y" THEN 140
210 IF S$ < > "N" AND S$ < > "n" THEN 18
0
220 TEXT : HOME : PRINT "Bye-bye"
230 END
240 REM INITIALIZE
250 REM TITLE
260 GOSUB 340
270 REM INSTRUCTIONS
280 GOSUB 390
290 REM VALUES
300 GOSUB 550
310 REM POSITION OF FINGERS
320 GOSUB 630
330 RETURN
340 REM TITLE
350 PRINT CHR$ (21): TEXT : HOME
360 VTAB 13: HTAB 15: PRINT "Mosaic Puzzle
370 FOR PAUSE = 1 TO 2000: NEXT
380 RETURN
390 REM INSTRUCTIONS
400 HOME
410 PRINT "Your goal is to rearrange the n
umbers
420 PRINT "of a 3x3 square so they look li
ke this"
430 VTAB 5: HTAB 15: INVERSE : PRINT "1";:
HTAB 20: PRINT "2";: HTAB 25: PRINT "
3"
440 VTAB 7: HTAB 15: PRINT "4";: HTAB 20: PRINT
"5";: HTAB 25: PRINT "6"
450 VTAB 9: HTAB 15: PRINT "7";: HTAB 20: PRINT
"8"; NORMAL
460 VTAB 13: HTAB 1: PRINT "Please enter a
number so that I can
470 PRINT "scramble your digits.
480 VTAB 16: HTAB 22: PRINT SPC( 20); CHR$
(7)
490 VTAB 16: HTAB 1: INVERSE : PRINT "SEED
(1 TO 1000)": NORMAL
500 INPUT " = ? ";S$
510 V = VAL (S$)
520 IF V < 1 OR V > 1000 THEN 480
530 Z = RND ( - V)
540 RETURN
550 REM VALUES
560 DIM R(3,3):D$ = CHR$ (4)

```

```

570 VTAB 19: HTAB 1: PRINT "Are you using
    a color monitor (Y/N) ? "; CHR$ (7);
580 GET S$
590 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 570
600 K1 = 15:K2 = 0:K3 = 15
610 IF S$ = "Y" OR S$ = "y" THEN K1 = 15:K
    2 = 5:K3 = 2
620 RETURN
630 REM FINGER
640 HOME
650 PRINT "When play begins, please positi
    on your
660 PRINT "fingers thusly for maximum ease
    !
670 INVERSE : VTAB 5: HTAB 10: PRINT "LEFT
    HAND": VTAB 11: HTAB 10: PRINT "RIGHT
    HAND": NORMAL
680 VTAB 7: HTAB 13: PRINT "D = middle fin
    ger
690 VTAB 8: HTAB 13: PRINT "F = index fing
    er
700 VTAB 13: HTAB 13: PRINT "L = middle fi
    nger
710 VTAB 14: HTAB 13: PRINT "M = index fin
    ger
720 VTAB 23: HTAB 14: PRINT "Press any key
    ";
730 GET S$
740 RETURN
750 REM GAME
760 REM CONTINUATION ?
770 GOSUB 910
780 REM READ OLD VALUES
790 IF S$ = "Y" OR S$ = "y" THEN GOSUB 97
    0
800 REM GENERATE NEW VALUES
810 IF S$ = "N" OR S$ = "n" THEN GOSUB 11
    00
820 REM FIND BLANK SQUARE
830 GOSUB 1410
840 REM DRAW BOARD
850 GOSUB 1470
860 REM LEGEND & INITIAL VALUES
870 GOSUB 2160
880 REM PLAY
890 GOSUB 2220
900 RETURN
910 REM CONTINUATION ?
920 TEXT : HOME

```

```

930  VTAB 1: HTAB 1: PRINT "Are you resumin
      g an old game (Y/N) ? "; CHR$ (7);
940  GET S$
950  IF S$ < > "Y" AND S$ < > "y" AND S$ <
      > "N" AND S$ < > "n" THEN 930
960  RETURN
970  REM OLD VALUES
980  VTAB 3: HTAB 1: PRINT "What's the name
      of your file"; CHR$ (7);
990  INPUT " ? ";FILE$
1000 IF FILE$ = "" THEN 980
1010 PRINT D$;"OPEN" + FILE$
1020 PRINT D$;"READ" + FILE$
1030 INPUT N
1040 FOR I = 1 TO 3
1050 FOR J = 1 TO 3
1060 INPUT R(I,J)
1070 NEXT J,I
1080 PRINT D$;"CLOSE" + FILE$
1090 RETURN
1100 REM NEW VALUES
1110 REM GENERATE DIGITS (0 TO 8)
1120 GOSUB 1170
1130 REM PLACE THEM IN R
1140 GOSUB 1330
1150 N = 0
1160 RETURN
1170 REM DIGITS
1180 FOR I = 0 TO 8:X(I) = I: NEXT
1190 T = 9
1200 FOR I = 1 TO 9
1210 P = INT (T * RND (1))
1220 Y(I) = X(P)
1230 REM STRIKE VALUE FROM X
1240 IF P < > T THEN GOSUB 1280
1250 T = T - 1
1260 NEXT I
1270 RETURN
1280 REM STRIKE VALUE
1290 FOR J = P TO T - 1
1300 X(J) = X(J + 1)
1310 NEXT J
1320 RETURN
1330 REM RECORD
1340 C = 1
1350 FOR I = 1 TO 3
1360 FOR J = 1 TO 3
1370 R(I,J) = Y(C)
1380 C = C + 1
1390 NEXT J,I

```

```

1400 RETURN
1410 REM BLANK SQUARE
1420 FOR I = 1 TO 3
1430 FOR J = 1 TO 3
1440 IF R(I,J) = 0 THEN RK = I:CK = J
1450 NEXT J,I
1460 RETURN
1470 REM DRAW BOARD
1480 REM BORDERS
1490 GOSUB 1550
1500 REM SQUARES
1510 GOSUB 1620
1520 REM NUMBERS
1530 GOSUB 1760
1540 RETURN
1550 REM BORDERS
1560 OR
1570 COLOR= K1
1580 FOR I = 0 TO 38
1590 HLIN 5,33 AT I
1600 NEXT
1610 RETURN
1620 REM SQUARES
1630 FOR R = 1 TO 3
1640 FOR C = 1 TO 3
1650 GOSUB 1680
1660 NEXT C,R
1670 RETURN
1680 REM COLOR SQUARES
1690 COLOR= K2
1700 X = C * 9 - 2
1710 Y = R * 12 - 10
1720 FOR L = Y TO Y + 9
1730 HLIN X,X + 6 AT L
1740 NEXT L
1750 RETURN
1760 REM NUMBERS
1770 COLOR= K3
1780 FOR R = 1 TO 3
1790 FOR C = 1 TO 3
1800 X = 9 * C - 1:Y = 12 * R - 3
1810 ON R(R,C) GOSUB 1840,1870,1910,1950,1
    990,2030,2070,2110
1820 NEXT C,R
1830 RETURN
1840 REM 1
1850 HLIN X + 1,X + 3 AT Y: VLIN Y - 6,Y AT
    X + 2: PLOT X + 1,Y - 5
1860 RETURN
1870 REM 2

```



```

1880 HLIN X,X + 4 AT Y: HLIN X + 1,X + 3 AT
    Y - 6: HLIN X + 2,X + 3 AT Y - 3
1890 PLOT X,Y - 5: PLOT X,Y - 1: PLOT X +
    1,Y - 2: VLIN Y - 5,Y - 4 AT X + 4
1900 RETURN
1910 REM 3
1920 HLIN X,X + 4 AT Y - 6: HLIN X + 1,X +
    3 AT Y: HLIN X + 2,X + 3 AT Y - 3
1930 PLOT X,Y - 1: PLOT X + 3,Y - 4: PLOT
    X + 4,Y - 5: VLIN Y - 2,Y - 1 AT X + 4
1940 RETURN
1950 REM 4
1960 HLIN X,X + 4 AT Y - 2: VLIN Y - 6,Y AT
    X + 3
1970 PLOT X,Y - 3: PLOT X + 1,Y - 4: PLOT
    X + 2,Y - 5
1980 RETURN
1990 REM 5
2000 HLIN X + 1,X + 3 AT Y: HLIN X,X + 3 AT
    Y - 4: HLIN X,X + 4 AT Y - 6: VLIN Y -
    3,Y - 1 AT X + 4
2010 PLOT X,Y - 5: PLOT X,Y - 1
2020 RETURN
2030 REM 6
2040 HLIN X + 1,X + 3 AT Y: HLIN X + 1,X +
    3 AT Y - 3: HLIN X + 2,X + 4 AT Y - 6
2050 VLIN Y - 4,Y - 1 AT X: VLIN Y - 2,Y -
    1 AT X + 4: PLOT X + 1,Y - 5
2060 RETURN
2070 REM 7
2080 HLIN X,X + 4 AT Y - 6: VLIN Y - 2,Y AT
    X + 1
2090 PLOT X + 2,Y - 3: PLOT X + 3,Y - 4: PLOT
    X + 4,Y - 5
2100 RETURN
2110 REM 8
2120 HLIN X + 1,X + 3 AT Y: HLIN X + 1,X +
    3 AT Y - 3: HLIN X + 1,X + 3 AT Y - 6
2130 VLIN Y - 5,Y - 4 AT X: VLIN Y - 5,Y -
    4 AT X + 4
2140 VLIN Y - 2,Y - 1 AT X: VLIN Y - 2,Y -
    1 AT X + 4
2150 RETURN
2160 REM LEGEND
2170 GAME$ = "ON"
2180 VTAB 22: HTAB 6: PRINT "D"; TAB( 13)"
    F"; TAB( 21)"L"; TAB( 26)"M"; TAB( 33)
    "Q=QUIT"
2190 INVERSE : VTAB 22: HTAB 7: PRINT "LEF
    T";: HTAB 14: PRINT "RIGHT";: HTAB 22:
    PRINT "UP";: HTAB 27: PRINT "DOWN"; NORMAL

```

```

2200 VTAB 24: HTAB 18: PRINT "MOVES = ";N;
2210 RETURN
2220 REM PLAY
2230 REM GET MOVE
2240 GOSUB 2350
2250 IF GAME$ = "QUIT" THEN 2340
2260 REM CHECK LEGALITY
2270 GOSUB 2460
2280 IF M$ = "ILLEGAL" THEN 2240
2290 REM MAKE MOVE
2300 GOSUB 2510
2310 REM CHECK FOR END OF GAME
2320 GOSUB 2710
2330 IF GAME$ = "ON" THEN 2240
2340 RETURN
2350 REM GET MOVE
2360 GET R$
2370 A = ASC (R$)
2380 IF A > 90 THEN A = A - 32
2390 IF A < > 68 AND A < > 70 AND A < >
76 AND A < > 77 AND A < > 81 THEN PRINT
CHR$ (7);: GOTO 2360
2400 IF A = 81 THEN GAME$ = "QUIT"
2410 IF A = 68 THEN DR$ = "LEFT"
2420 IF A = 70 THEN DR$ = "RIGHT"
2430 IF A = 76 THEN DR$ = "UP"
2440 IF A = 77 THEN DR$ = "DOWN"
2450 RETURN
2460 REM CHECK LEGALITY
2470 M$ = ""
2480 IF (DR$ = "LEFT" AND CK = 3) OR (DR$ =
"RIGHT" AND CK = 1) OR (DR$ = "UP" AND
RK = 3) OR (DR$ = "DOWN" AND RK = 1) THEN
M$ = "ILLEGAL"
2490 IF M$ < > "" THEN FOR S = 1 TO 20:S
OUND = PEEK ( - 16336): NEXT
2500 RETURN
2510 REM MAKE MOVE
2520 REM IDENTIFY SQUARE TO MOVE
2530 GOSUB 2650
2540 REM BLANK OUT SQUARE
2550 R = RM:CM = CM: GOSUB 1680
2560 REM FILL IN SQUARE
2570 X = 9 * CK - 1:Y = 12 * RK - 3: COLOR=
K3
2580 ON R(RM,CM) GOSUB 1840,1870,1910,1950
,1990,2030,2070,2110
2590 REM UPDATE VALUES
2600 HD = R(RK,CK):R(RK,CK) = R(RM,CM):R(RM
,CM) = HD

```

```

2610 RK = RM:CK = CM
2620 N = N + 1
2630 VTAB 24: HTAB 26: PRINT N;
2640 RETURN
2650 REM IDENTIFY SQUARE TO MOVE
2660 IF DR$ = "LEFT" THEN RM = RK:CM = CK +
1
2670 IF DR$ = "RIGHT" THEN RM = RK:CM = CK
- 1
2680 IF DR$ = "UP" THEN RM = RK + 1:CM = C
K
2690 IF DR$ = "DOWN" THEN RM = RK - 1:CM =
CK
2700 RETURN
2710 REM CHECK FOR END
2720 GAME$ = "OVER"
2730 V = 1
2740 FOR I = 1 TO 2
2750 FOR J = 1 TO 3
2760 IF R(I,J) < > V THEN GAME$ = "ON"
2770 V = V + 1
2780 NEXT J,I
2790 IF R(3,1) < > 7 OR R(3,2) < > 8 THEN
GAME$ = "ON"
2800 RETURN
2810 REM QUIT
2820 VTAB 22: HTAB 6: PRINT SPC( 34)
2830 VTAB 24: HTAB 18: PRINT SPC( 13)
2840 VTAB 22: HTAB 4: PRINT "Should I save
your board (Y/N) ? "; CHR$(7);
2850 GET S$
2860 IF S$ < > "Y" AND S$ < > "y" AND S$
< > "N" AND S$ < > "n" THEN 2840
2870 VTAB 22: HTAB 4: PRINT SPC( 36)
2880 IF S$ = "Y" OR S$ = "y" THEN GOSUB 2
900
2890 RETURN
2900 REM SAVE GAME
2910 VTAB 22: HTAB 1: PRINT "File Name"; CHR$(
7);
2920 INPUT " = ? ";FILE$
2930 IF FILE$ = "" THEN 2910
2940 PRINT D$;"OPEN" + FILE$
2950 PRINT D$;"WRITE" + FILE$
2960 PRINT N
2970 FOR I = 1 TO 3
2980 FOR J = 1 TO 3
2990 PRINT R(I,J)
3000 NEXT J,I

```

```

3010 PRINT D$;"CLOSE" + FILE$
3020 VTAB 22; HTAB 1; PRINT FILE$;" is suc
cessfully saved.
3030 RETURN
3040 REM END OF GAME
3050 VTAB 22; HTAB 6; PRINT SPC( 34)
3060 VTAB 24; HTAB 18; PRINT SPC( 13)
3070 VTAB 24; HTAB 13; FLASH ; PRINT "THE
GAME'S OVER !"; NORMAL
3080 FOR PAUSE = 1 TO 30; PRINT CHR$ (7);
: NEXT
3090 VTAB 24; HTAB 13; PRINT SPC( 17);
3100 VTAB 22; HTAB 6; PRINT "You used ";N;
" moves."
3110 RETURN

```

Spelling Quiz

"Spelling Quiz" lets you enter your own word list, displays some words correctly spelled, others incorrectly. It even gives you hints if you misspell a word.

"Spelling Quiz" is designed to help all of us spell better. Everyone, from grade schoolers to grownups, can use some spelling practice at one time or another. With this program, you use your own list of up to 25 words. The computer not only asks you to spell the words correctly, but it will often display an incorrectly spelled version of the word. If you enter a wrong spelling, the program even gives you a hint of how to get it right.

Word Files

Once you've typed in and saved Spelling Quiz, run it. An instruction screen appears and a prompt reminds you to press the CAPS LOCK key, applicable only to the Apple IIc. Next, you'll see a list of instructions. Enter a seed value as instructed. After this, you will see a menu. If this is your first time using Spelling Quiz, you need to press the 1 key so that you can create a file of words. The screen changes and you can begin entering your words. They might be words you're studying in school, words you have trouble spelling, or perhaps words you want to add to your spelling vocabulary.

Up to 25 words can be stored in a single file. Since you can create multiple files in Spelling Quiz, you can practice spelling as many words as you want. When you've completed your list (it doesn't have to be 25 words; it can be less), hit the RETURN key instead of typing in a word. Select a filename for this list and type it in. When you hit the RETURN key, it's saved to disk automatically.

As an example list, type in the following words. Make sure you enter them correctly.

ASYNCHRONOUS
BALALAIKA
CHASSIS

DOUGHNUT
EMPHYSEMA
FLAMINGO

GARGLE	QUIXOTIC
HONEYSUCKLE	RENAISSANCE
IRELAND	SANGUINE
JAVELIN	TRANSCENDENTAL
KNOWLEDGEABLE	UNIVERSAL
LEPRECHAUN	VERBIAGE
MAHOGANY	WROUGHT
NIGHTINGALE	YIDDISH
OMNISCIENT	ZOOLOGY
PAROXYSM	

Save the list to disk, using a filename such as LIST1. Once it's saved, the program returns to the original menu, and you can press the 2 key to practice your spelling. Enter the filename LIST1 and it loads into the computer. The screen changes and you'll see the computer-spelled version of the first word from your list. It might be spelled AYSNCHRONOUS.

You get one chance to spell the word correctly. Perhaps you think AYSNCHRONOUS is correct, and type it in. The computer indicates that's incorrect and tells you the correct spelling.

You may want to take turns with friends using the same list of words. Or try entering different lists to stump each other. In either case, the player with the highest percentage wins the Spelling Quiz.

Spelling Quiz

```

100 REM SPELLING QUIZ
110 REM INITIALIZE
120 GOSUB 180
130 REM SELECT FROM MENU
140 GOSUB 830
150 ON PICK GOSUB 950,1340
160 IF PICK < > 3 THEN 140
170 END
180 REM INITIALIZE
190 REM TITLE
200 GOSUB 260
210 REM INSTRUCTIONS
220 GOSUB 360
230 REM MISSPELLINGS
240 GOSUB 630
250 RETURN
260 REM TITLE
270 PRINT CHR$(21): TEXT : HOME
280 VTAB 13: HTAB 14: PRINT "Spelling Quiz
    
```

```

290 FOR PAUSE = 1 TO 2500: NEXT
300 REM MAXIMUM NUMBERS OF WORDS & LETTER
    REPLACEMENTS
310 DATA 25,20
320 READ X,Y
330 DIM W$(X),WF$(Y,4),WT$(Y,4)
340 D$ = CHR$(4)
350 RETURN
360 REM INSTRUCTIONS
370 HOME
380 PRINT "To improve your spelling, first
    store
390 PRINT "up to ";X;" words on file. Reload
    ad
400 PRINT "them at your leisure.
410 PRINT
420 PRINT "I'll display each of the words
    in turn,"
430 PRINT "but not always correctly.
440 PRINT
450 PRINT "If you think a word is okay, just key
    st key
460 PRINT "it back in, as is.
470 PRINT
480 PRINT "If not, enter what you think is
    the
490 PRINT "correct spelling.
500 PRINT
510 PRINT "Please set the "; INVERSE : PRINT
    "CAPS LOCK";: NORMAL : PRINT " key to
520 PRINT "uppercase before continuing.
530 PRINT
540 PRINT "And please enter a number so that
    at I can"
550 PRINT "misspell your words randomly.
560 VTAB 20: HTAB 22: PRINT SPC(20); CHR$(
    7)
570 VTAB 20: HTAB 1: INVERSE : PRINT "SEED
    (1 TO 1000)";: NORMAL
580 INPUT " = ? ";S$
590 V = VAL(S$)
600 IF V < 1 OR V > 1000 THEN 560
610 Z = RND(-V)
620 RETURN
630 REM MISSPELLINGS
640 REM NUMBER OF STRING REPLACEMENTS: 1,
    2,3, & 4 LETTER
650 DATA 6,15,4,3
660 FOR I = 1 TO 4
670 READ P(I)

```

```

680 NEXT
690 REM ONE LETTER (FROM,TO)
700 DATA Z,S,J,G,O,U,U,O,E,I,I,E
710 REM TWO LETTER
720 DATA IE,EI,EI,IE,SY,YS,GH,HG,TH,HT,LE
,AL,AL,EL,EL,AL,AU,AW,OU,OW,SC,SK,QU,K
W,CA,KA,CU,KU,CO,KO
730 REM THREE LETTER
740 DATA PHY,PHE,ITY,ETY,EGE,OGE,ILY,ELY
750 REM FOUR LETTER
760 DATA TION,SHUN,MENT,MANT,IBLE,ABLE
770 REM READ
780 FOR I = 1 TO 4
790 FOR J = 1 TO P(I)
800 READ WF$(J,I),WT$(J,I)
810 NEXT J,I
820 RETURN
830 REM MENU
840 HOME
850 VTAB 7: HTAB 10: PRINT "Would you like
to
860 INVERSE : FOR I = 1 TO 3: VTAB 2 * I +
7: HTAB 10: PRINT I: NEXT : NORMAL
870 VTAB 9: HTAB 12: PRINT "Create a file
of words
880 VTAB 11: HTAB 12: PRINT "Test your spe
lling
890 VTAB 13: HTAB 12: PRINT "Exit
900 VTAB 15: HTAB 10: PRINT "Choice = ? ";
CHR$(7);
910 GET S$
920 PICK = VAL (S$)
930 IF PICK < 1 OR PICK > 3 THEN 900
940 RETURN
950 REM CREATE FILE
960 REM ENTER DATA
970 GOSUB 1030
980 REM ENTER FILE NAME
990 GOSUB 1190
1000 REM WRITE
1010 GOSUB 1250
1020 RETURN
1030 REM DATA
1040 HOME
1050 PRINT "Please enter your words. Hit "
;: INVERSE : PRINT "RETURN"; NORMAL
1060 PRINT "when through.
1070 N = X
1080 FOR I = 1 TO X
1090 VTAB 5: HTAB 16: PRINT SPC(40); CHR$(
7)

```



```

1100 VTAB 5: HTAB 1: PRINT "Word No. ";I; TAB(
12)"= ";
1110 INPUT "? ";W$: VTAB 23: HTAB 11: PRINT
    SPC( 17)
1120 IF LEN (W$) > 0 AND LEN (W$) < 3 THEN
    VTAB 23: HTAB 11: PRINT "That's too e
asy !": GOTO 1090
1130 IF LEN (W$) > 25 THEN VTAB 23: HTAB
    11: PRINT "That's too hard !": GOTO 10
90
1140 W$(I) = W$
1150 IF W$ = "" THEN N = I - 1:I = X
1160 NEXT
1170 IF N = 0 THEN 1070
1180 RETURN
1190 REM FILE NAME
1200 HOME
1210 VTAB 1: HTAB 1: PRINT "What's your fi
le name "; CHR$ (7);
1220 INPUT "? ";FILE$
1230 IF FILE$ = "" THEN 1210
1240 RETURN
1250 REM WRITE
1260 PRINT D$;"OPEN" + FILE$
1270 PRINT D$;"WRITE" + FILE$
1280 PRINT N
1290 FOR I = 1 TO N
1300 PRINT W$(I)
1310 NEXT
1320 PRINT D$;"CLOSE" + FILE$
1330 RETURN
1340 REM TEST SPELLING
1350 REM READ FILE NAME
1360 GOSUB 1190
1370 REM READ DATA
1380 GOSUB 1440
1390 REM QUIZ
1400 GOSUB 1530
1410 REM DISPLAY SCORE
1420 GOSUB 3020
1430 RETURN
1440 REM READ DATA
1450 PRINT D$;"OPEN" + FILE$
1460 PRINT D$;"READ" + FILE$
1470 INPUT N
1480 FOR I = 1 TO N
1490 INPUT W$(I)
1500 NEXT
1510 PRINT D$;"CLOSE" + FILE$
1520 RETURN

```

```

1530 REM QUIZ
1540 REM DRAW BOARD
1550 GOSUB 1700
1560 REM LABEL
1570 GOSUB 1840
1580 REM TEST
1590 FOR I = 1 TO N
1600 REM SCRAMBLE WORD
1610 GOSUB 1940
1620 REM DISPLAY IT
1630 GOSUB 2760
1640 REM ENTER GUESS
1650 GOSUB 2800
1660 REM EVALUATE IT
1670 GOSUB 2850
1680 NEXT I
1690 RETURN
1700 REM BOARD
1710 HOME : INVERSE :BK$ = CHR$ (32)
1720 FOR L = 1 TO 39
1730 VTAB 1: HTAB L: PRINT BK$
1740 VTAB 2: HTAB L: PRINT BK$
1750 VTAB 3: HTAB L: PRINT BK$
1760 VTAB 24: HTAB L: PRINT BK$;
1770 NEXT L
1780 FOR L = 4 TO 23
1790 VTAB L: HTAB 1: PRINT BK$
1800 VTAB L: HTAB 39: PRINT BK$
1810 NEXT L
1820 NORMAL
1830 RETURN
1840 REM LABEL
1850 INVERSE
1860 VTAB 11: HTAB 7: PRINT "WORD:"
1870 VTAB 13: HTAB 3: PRINT "SPELLING:"
1880 NORMAL
1890 VTAB 2: HTAB 13: PRINT " SPELLING QUI
Z "
1900 VTAB 5: HTAB 17: PRINT "Number of Wor
ds = ";N
1910 VTAB 6: HTAB 20: PRINT "Number Right
= 0"
1920 NR = 0
1930 RETURN
1940 REM SCRAMBLE WORD
1950 REM USER READY
1960 GOSUB 2000
1970 REM SCRAMBLE
1980 GOSUB 2120
1990 RETURN

```

```

2000 REM USER READY
2010 VTAB 22: HTAB 17: PRINT SPC( 7)
2020 VTAB 22: HTAB 6: PRINT "Press "; INVERSE
    : PRINT "RETURN"; NORMAL : PRINT " fo
    r Word No. ";I;" "; CHR$( 7);
2030 GET S$
2040 IF ASC (S$) < > 13 THEN 2020
2050 VTAB 22: HTAB 6: PRINT SPC( 30)
2060 VTAB 22: HTAB 15: FLASH : PRINT "SCRA
    MBLING": NORMAL
2070 VTAB 11: HTAB 13: PRINT SPC( 25)
2080 VTAB 13: HTAB 13: PRINT SPC( 25)
2090 VTAB 15: HTAB 3: PRINT SPC( 35)
2100 FOR PAUSE = 1 TO 30: SOUND = PEEK ( -
    16336): FOR TIME = 1 TO 60: NEXT TIME,
    PAUSE
2110 RETURN
2120 REM SCRAMBLE (TRY 4 TIMES)
2130 W$ = W$(I): LN = LEN (W$): JOB$ = "": L$
    = "": R$ = ""
2140 FOR Q = 1 TO 4
2150 V = INT (5 * RND (1)) + 1
2160 ON V GOSUB 2200, 2320, 2440, 2510, 2630
2170 IF JOB$ = "DONE" THEN Q = 4
2180 NEXT Q
2190 RETURN
2200 REM ONE LETTER
2210 FOR J = 1 TO P(1)
2220 FOR L = 1 TO LN
2230 IF MID$( W$,L,1) = WF$(J,1) THEN S$ =
    WT$(J,1): GOSUB 2260: L = LN: J = P(1)
2240 NEXT L, J
2250 RETURN
2260 REM REPLACE 1 LETTER WITH ANOTHER
2270 IF L < > 1 THEN L$ = LEFT$( W$,L -
    1)
2280 IF L < > LN THEN R$ = MID$( W$,L +
    1)
2290 W$ = L$ + S$ + R$
2300 JOB$ = "DONE"
2310 RETURN
2320 REM TWO LETTERS
2330 FOR J = 1 TO P(2)
2340 FOR L = 1 TO LN - 1
2350 IF MID$( W$,L,2) = WF$(J,2) THEN S$ =
    WT$(J,2): GOSUB 2380: L = LN - 1: J = P(
    2)
2360 NEXT L, J
2370 RETURN

```

```

2380 REM REPLACE 2 LETTERS WITH 1 OR 2 NE
    W ONES
2390 IF L < > 1 THEN L$ = LEFT$ (W$,L -
    1)
2400 IF L < > LN - 1 THEN R$ = MID$ (W$,
    L + 2)
2410 W$ = L$ + S$ + R$
2420 JOB$ = "DONE"
2430 RETURN
2440 REM DOUBLE LETTERS
2450 FOR J = 65 TO 90
2460 C$ = CHR$ (J) + CHR$ (J)
2470 FOR L = 1 TO LN - 1
2480 IF MID$ (W$,L,2) = C$ THEN S$ = CHR$
    (J): GOSUB 2380:L = LN - 1:J = 90
2490 NEXT L,J
2500 RETURN
2510 REM THREE LETTERS
2520 FOR J = 1 TO P(3)
2530 FOR L = 1 TO LN - 2
2540 IF MID$ (W$,L,3) = WF$(J,3) THEN S$ =
    WT$(J,3): GOSUB 2570:L = LN - 2:J = P(
    3)
2550 NEXT L,J
2560 RETURN
2570 REM REPLACE 3 LETTERS
2580 IF L < > 1 THEN L$ = LEFT$ (W$,L -
    1)
2590 IF L < > LN - 2 THEN R$ = MID$ (W$,
    L + 3)
2600 W$ = L$ + S$ + R$
2610 JOB$ = "DONE"
2620 RETURN
2630 REM FOUR LETTERS
2640 IF LN < 5 THEN 2690
2650 FOR J = 1 TO P(4)
2660 FOR L = 1 TO LN - 3
2670 IF MID$ (W$,L,4) = WF$(J,4) THEN S$ =
    WT$(J,4): GOSUB 2700:L = LN - 3:J = P(
    4)
2680 NEXT L,J
2690 RETURN
2700 REM REPLACE 4 LETTERS
2710 IF L < > 1 THEN L$ = LEFT$ (W$,L -
    1)
2720 IF L < > LN - 3 THEN R$ = MID$ (W$,
    L + 4)
2730 W$ = L$ + S$ + R$
2740 JOB$ = "DONE"
2750 RETURN

```

```

2760 REM DISPLAY WORD
2770 VTAB 22: HTAB 15: PRINT SPC( 10)
2780 VTAB 11: HTAB 13: PRINT W$
2790 RETURN
2800 REM GUESS
2810 VTAB 13: HTAB 13: PRINT CHR$( 7);: HTAB
12: INPUT " ";G$
2820 VTAB 13: HTAB 39: INVERSE : PRINT CHR$
(32): NORMAL
2830 IF G$ = "" THEN 2810
2840 RETURN
2850 REM EVALUATE ANSWER
2860 R$ = "WRONG !"
2870 IF G$ = W$(I) THEN R$ = "RIGHT !":NR =
NR + 1
2880 REM FLASH RESULT
2890 GOSUB 2940
2900 IF R$ = "RIGHT !" THEN VTAB 6: HTAB
35: PRINT NR
2910 REM DISPLAY RIGHT ANSWER
2920 IF R$ = "WRONG !" THEN GOSUB 2980
2930 RETURN
2940 REM RESULT
2950 VTAB 22: HTAB 17: FLASH : PRINT R$: NORMAL

2960 FOR PAUSE = 1 TO 20: PRINT CHR$( 7);
: NEXT PAUSE
2970 RETURN
2980 REM RIGHT ANSWER
2990 VTAB 15: HTAB 5: INVERSE : PRINT "ANS
WER: ";: NORMAL
3000 PRINT " ";W$(I)
3010 RETURN
3020 REM SCORE
3030 PCT = INT (100 * NR / N)
3040 IF PCT > 85 THEN T$ = "Very Good !"
3050 IF PCT > = 50 AND PCT < = 85 THEN T
$ = "You're Making Progress !"
3060 IF PCT < 50 THEN T$ = "Let's try thos
e words again !"
3070 VTAB 22: HTAB 17: PRINT SPC( 7)
3080 VTAB 20: HTAB (20 - LEN (T$) / 2): PRINT
T$
3090 VTAB 22: HTAB 14: PRINT "Press any ke
y ";
3100 GET S$
3110 RETURN

```

Hi-Q

"Hi-Q," a famous European solitaire game of finesse and foresight, tests your planning and logic. Thirty-two pegs must be moved and jumped so that only one remains. Unfortunately, it's not as simple as it sounds.

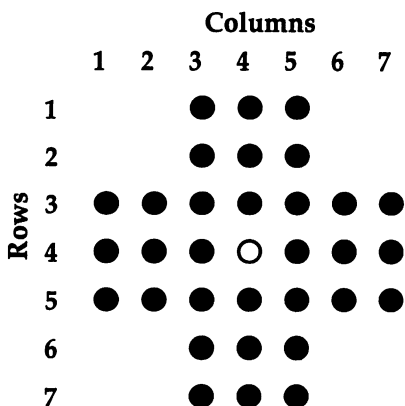
"Hi-Q" is a peg game of thought and skillful movement. Thirty-two pegs appear on a cross-shaped board, with only the center position empty. By jumping one peg over another, a piece is removed from the board. Your goal? To remove as many pegs as possible.

You get a perfect score in Hi-Q if only one peg remains. It is somewhat like pitching a shutout. The ultimate thrill is leaving the one peg in the center of the board; that's like a no-hitter.

Leapfrog

Enter and save a copy of Hi-Q, then run the program. After an instruction screen and color monitor prompt, the board is drawn on the screen. A stylized illustration of the board is shown below in Figure 1.

Figure 1. Hi-Q Game Board



You remove a peg from the board by jumping it with another peg. In fact, every move in Hi-Q must be a jump. Only horizontal and vertical moves are allowed. Just follow the

computer's prompts, entering the row and column numbers of the peg to move, and then the coordinates of its destination. It's as simple as that. Figure 2 shows how peg 4,2 (row 4, column 2) would be moved to the hole at coordinates 4,4.

Figure 2. Peg Jumping

Before							After						
1	2	3	4	5	6	7	1	2	3	4	5	6	7
1			●	●	●		1		●	●	●		
2			●	●	●		2		●	●	●		
3	●	●	●	●	●	●	3	●	●	●	●	●	●
4	●	●	●	○	●	●	4	●	○	○	●	●	●
5	●	●	●	●	●	●	5	●	●	●	●	●	●
6			●	●	●		6		●	●	●		
7			●	●	●		7		●	●	●		

When peg 4, 2 is moved to hole 4,4 locations 4,2 and 4,3 become vacant. Peg 4,3 has been jumped and consequently removed from play. A total of 31 pegs remain.

Now, locations 4,2 and 4,3 are empty. Since the peg at 4,3 was jumped over, it's removed from the board. Thirty-one pegs remain.

If you change your mind in the middle of a move, don't fret. Simply enter a destination that's illegal. A hole on the other side of the board will work nicely. The computer will flash a warning and then return you to the *Peg to Move: Row = ?* prompt. There's no penalty for doing this. If you mistakenly request an occupied location, you'll see a message to that effect. Enter the correct coordinates and the game continues.

Technical note. If you have an Apple II or II+ with little memory, you may encounter an OUT OF MEMORY error when you run this program. If so, eliminate lines 100, 430-480, 1070-1130, and 3160-3260. This will solve the problem.

HI-Q

```

100 LOMEM: 16384
110 REM HI-Q
120 REM INITIALIZE
130 GOSUB 230
140 REM PLAY GAME
150 GOSUB 550
160 REM PLAY AGAIN
170 VTAB 24: HTAB 12: PRINT "Play again (Y
/N) ? "; CHR$ (7);
180 GET S$
190 IF S$ = "Y" OR S$ = "y" THEN 150
200 IF S$ < > "N" AND S$ < > "n" THEN 17
0
210 TEXT : HOME : PRINT "Bye-bye"
220 END
230 REM INITIALIZE
240 REM TITLE
250 GOSUB 360
260 REM INSTRUCTIONS
270 GOSUB 410
280 K = 15
290 DIM R(7,7),C1(K),C2(K)
300 REM CIRCLE POINTS
310 DATA 5,9,3,11,2,12,1,13,1,13,0,14,0,1
4,0,14,0,14,0,14,1,13,1,13,2,12,3,11,5
,9
320 FOR I = 1 TO K
330 READ C1(I),C2(I)
340 NEXT
350 RETURN
360 REM TITLE
370 PRINT CHR$ (21): TEXT : HOME
380 VTAB 12: HTAB 18: PRINT "Hi-Q
390 FOR PAUSE = 1 TO 2000: NEXT
400 RETURN
410 REM INSTRUCTIONS
420 HOME
430 PRINT "I'm about to place 32 pegs on a
cross-"
440 PRINT "shaped board, with the center e
mpty.
450 PRINT
460 PRINT "Try to remove as many pegs as p
ossible,"
470 PRINT "with a peg lifted from the boar
d when
480 PRINT "it's jumped.

```



```

490 VTAB 11: HTAB 1: PRINT "Are you using
    a color monitor (Y/N) ? "; CHR$ (7);
500 GET S$
510 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 490
520 K1 = 0:K2 = 3:K3 = 3
530 IF S$ = "Y" OR S$ = "y" THEN K1 = 6:K2
    = 3:K3 = 5
540 RETURN
550 REM PLAY GAME
560 REM INITIAL POSITION
570 GOSUB 680
580 REM ENTER MOVE
590 GOSUB 1380
600 REM MAKE MOVE
610 GOSUB 2190
620 REM CHECK FOR END OF GAME
630 GOSUB 2460
640 IF GAME$ = "ON" THEN 590
650 REM END
660 GOSUB 2690
670 RETURN
680 REM INITIAL POSITION
690 REM RECORD INITIAL POSITION
700 GOSUB 780
710 REM LABEL BOARD
720 GOSUB 890
730 REM DRAW PEGS
740 GOSUB 1150
750 REM DRAW LINES
760 GOSUB 1290
770 RETURN
780 REM RECORD
790 REM 1=FILLED; 0=VACANT;-9=OFF BOARD
800 FOR R = 1 TO 7
810 FOR C = 1 TO 7
820 R(R,C) = 1
830 IF (R < 3 AND C < 3) OR (R < 3 AND C >
    5) OR (R > 5 AND C < 3) OR (R > 5 AND
    C > 5) THEN R(R,C) = - 9
840 NEXT C,R
850 REM CENTER
860 R(4,4) = 0
870 PEGS = 32
880 RETURN
890 REM LABEL BOARD
900 HOME
910 HGR : HCOLOR= K1: HPLLOT 1,1: CALL 6245
    4
920 HCOLOR= 0: HPLLOT 0,159 TO 279,159

```

```

930 REM COLUMNS
940 FOR I = 1 TO 7
950 X = 30 * I + 20:Y = 10
960 GOSUB 2810
970 HCOLOR= 3
980 ON I GOSUB 2870,2900,2950,2990,3030,30
    70,3120
990 NEXT
1000 REM ROWS
1010 FOR I = 1 TO 7
1020 X = 24:Y = 20 * I + 10
1030 GOSUB 2810
1040 HCOLOR= 3
1050 ON I GOSUB 2870,2900,2950,2990,3030,3
    070,3120
1060 NEXT
1070 REM LETTERS
1080 FOR I = 1 TO 3
1090 X = 7:Y = 10 * I + 65
1100 GOSUB 2810
1110 HCOLOR= 3
1120 ON I GOSUB 3160,3200,3230
1130 NEXT
1140 RETURN
1150 REM DRAW PEGS
1160 HCOLOR= K3
1170 FOR R = 1 TO 7
1180 FOR C = 1 TO 7
1190 IF R(R,C) = 1 THEN GOSUB 1220
1200 NEXT C,R
1210 RETURN
1220 REM PEGS
1230 X = C * 30 + 15:Y = R * 20
1240 FOR I = 1 TO K
1250 HPLOT X + C1(I),Y TO X + C2(I),Y
1260 Y = Y + 1
1270 NEXT I
1280 RETURN
1290 REM LINES
1300 HCOLOR= 3
1310 FOR I = 1 TO 2
1320 HPLLOT 45,50 + I TO 95,50 + I: HPLLOT 1
    89,50 + I TO 239,50 + I
1330 HPLLOT 45,121 + I TO 95,121 + I: HPLLOT
    189,121 + I TO 239,121 + I
1340 HPLLOT 93 + I,20 TO 93 + I,52: HPLLOT 9
    3 + I,122 TO 93 + I,154
1350 HPLLOT 188 + I,20 TO 188 + I,52: HPLLOT
    188 + I,122 TO 188 + I,154

```

```

1360 NEXT I
1370 RETURN
1380 REM ENTER MOVE
1390 REM PEG TO MOVE
1400 GOSUB 1490
1410 REM MAKE IT SQUARE
1420 GOSUB 1750
1430 REM HOLE TO FILL
1440 GOSUB 1880
1450 REM CHECK FOR LEGAL MOVE
1460 GOSUB 1960
1470 IF R$ = "ILLEGAL" THEN 1400
1480 RETURN
1490 REM PEG TO MOVE
1500 A$ = "Peg to move: "
1510 GOSUB 1570
1520 REM MAKE SURE PEG IS THERE
1530 VTAB 23: HTAB 15: PRINT SPC( 15)
1540 IF R(R,C) = 0 THEN VTAB 23: HTAB 15:
    INVERSE : PRINT "EMPTY SPACE !": NORMAL
    : GOTO 1510
1550 R1 = R: C1 = C
1560 RETURN
1570 REM CHOOSE ROW & COLUMN
1580 VTAB 21: HTAB 1: PRINT SPC( 40)
1590 REM ROW
1600 VTAB 21: HTAB 1: PRINT A$;"Row = ? ";
    CHR$( 7);
1610 GET S$
1620 R = VAL (S$)
1630 IF R < 1 OR R > 7 THEN 1600
1640 PRINT R
1650 REM COLUMN
1660 VTAB 21: HTAB 25: PRINT "Column = ? "
    ; CHR$( 7);
1670 GET S$
1680 C = VAL (S$)
1690 IF C < 1 OR C > 7 THEN 1660
1700 PRINT C
1710 REM CHECK FOR MOVE ON BOARD
1720 VTAB 23: HTAB 15: PRINT SPC( 15)
1730 IF R(R,C) = - 9 THEN VTAB 23: HTAB
    16: INVERSE : PRINT "OFF BOARD !": NORMAL
    : GOTO 1580
1740 RETURN
1750 REM MAKE IT SQUARE
1760 REM ERASE CIRCLE
1770 HCOLOR= K1: GOSUB 1220
1780 REM DRAW SQUARE
1790 HCOLOR= 3

```

```

1800 GOSUB 1820
1810 RETURN
1820 REM SQUARE
1830 X = C * 30 + 18:Y = R * 20 + 3
1840 FOR I = 0 TO 7
1850 HPLLOT X + I,Y TO X + I,Y + 8
1860 NEXT I
1870 RETURN
1880 REM HOLE TO FILL
1890 A$ = "Hole to fill: "
1900 GOSUB 1570
1910 REM MAKE SURE HOLE IS THERE
1920 VTAB 23: HTAB 15: PRINT SPC( 15)
1930 IF R(R,C) < > 0 THEN VTAB 23: HTAB
17: INVERSE : PRINT "NO HOLE !": NORMAL
: GOTO 1900
1940 R2 = R:C2 = C
1950 RETURN
1960 REM CHECK LEGALITY
1970 ROW = 0:COL = 0
1980 R$ = "ILLEGAL"
1990 REM HORIZONTAL
2000 MAX = C2
2010 IF C1 > C2 THEN MAX = C1
2020 COL = MAX - 1
2030 IF R1 = R2 AND ABS (C1 - C2) = 2 AND
R(R1, COL) = 1 THEN R$ = "LEGAL"
2040 REM VERTICAL
2050 MAX = R2
2060 IF R1 > R2 THEN MAX = R1
2070 ROW = MAX - 1
2080 IF C1 = C2 AND ABS (R1 - R2) = 2 AND
R(ROW,C1) = 1 THEN R$ = "LEGAL"
2090 IF R$ = "ILLEGAL" THEN GOSUB 2110
2100 RETURN
2110 REM ILLEGAL MOVE
2120 VTAB 23: HTAB 15: FLASH : PRINT "ILLE
GAL MOVE !": NORMAL
2130 FOR PAUSE = 1 TO 500:SOUND = PEEK ( -
16336): NEXT
2140 REM REDRAW PEG
2150 HCOLOR= K3
2160 R = R1:C = C1: GOSUB 1220
2170 VTAB 23: HTAB 1: PRINT SPC( 40)
2180 RETURN
2190 REM MAKE MOVE
2200 REM ERASE SQUARE
2210 HCOLOR= K1:R = R1:C = C1: GOSUB 1820
2220 REM DRAW NEW PEG
2230 HCOLOR= K3:R = R2:C = C2: GOSUB 1220

```

```

2240 REM ERASE MIDDLE PEG
2250 HCOLOR= K1
2260 IF R1 = R2 THEN GOSUB 2320
2270 IF R1 < > R2 THEN GOSUB 2390
2280 PEGS = PEGS - 1
2290 R(R1,C1) = 0
2300 R(R2,C2) = 1
2310 RETURN
2320 REM HORIZONTAL MOVE
2330 MAX = C2
2340 IF C1 > C2 THEN MAX = C1
2350 COL = MAX - 1
2360 R = R1;C = COL: GOSUB 1220
2370 R(R1,COL) = 0
2380 RETURN
2390 REM VERTICAL MOVE
2400 MAX = R2
2410 IF R1 > R2 THEN MAX = R1
2420 ROW = MAX - 1
2430 R = ROW;C = C1: GOSUB 1220
2440 R(ROW,C1) = 0
2450 RETURN
2460 REM CHECK FOR MORE MOVES
2470 GAME$ = "OVER"
2480 REM HORIZONTAL
2490 GOSUB 2530
2500 REM VERTICAL
2510 IF GAME$ = "OVER" THEN GOSUB 2610
2520 RETURN
2530 REM HORIZONTAL
2540 FOR I = 1 TO 7
2550 FOR J = 1 TO 5
2560 IF (I < 3 AND J < > 3) OR (I > 5 AND
J < > 3) THEN 2590
2570 IF R(I,J) = 1 AND R(I,J + 1) = 1 AND
R(I,J + 2) = 0 THEN GAME$ = "ON":J = 5
:I = 7
2580 IF R(I,J) = 0 AND R(I,J + 1) = 1 AND
R(I,J + 2) = 1 THEN GAME$ = "ON":J = 5
:I = 7
2590 NEXT J,I
2600 RETURN
2610 REM VERTICAL
2620 FOR I = 1 TO 7
2630 FOR J = 1 TO 5
2640 IF (I < 3 AND J < > 3) OR (I > 5 AND
J < > 3) THEN 2670
2650 IF R(J,I) = 1 AND R(J + 1,I) = 1 AND
R(J + 2,I) = 0 THEN GAME$ = "ON":J = 5
:I = 7

```

```

2660 IF R(J,I) = 0 AND R(J + 1,I) = 1 AND
    R(J + 2,I) = 1 THEN GAME$ = "ON";J = 5
    :I = 7
2670 NEXT J,I
2680 RETURN
2690 REM OVER
2700 VTAB 21: HTAB 1: PRINT SPC( 40)
2710 VTAB 23: HTAB 13: FLASH : PRINT "THE
    GAME'S OVER !": NORMAL
2720 FOR PAUSE = 1 TO 30: PRINT CHR$ (7);
    : NEXT
2730 P$ = " pegs remain. "
2740 IF PEGS = 1 THEN RK$ = "You're a geni
    us !":P$ = " peg remains. "
2750 IF PEGS = 2 OR PEGS = 3 THEN RK$ = "O
    utstanding !"
2760 IF PEGS = 4 OR PEGS = 5 THEN RK$ = "N
    ot bad."
2770 IF PEGS > 5 THEN RK$ = "Turkey !"
2780 VTAB 23: HTAB 13: PRINT SPC( 17)
2790 VTAB 21: HTAB 6: PRINT PEGS;P$;RK$
2800 RETURN
2810 REM BLACK BACKGROUND FOR LETTER
2820 HCOLOR= 0:X1 = X - 1:Y1 = Y + 1
2830 FOR L = X1 TO X1 + 6
2840 HPLLOT L,Y1 TO L,Y1 - 8
2850 NEXT L
2860 RETURN
2870 REM 1
2880 HPLLOT X + 1,Y TO X + 3,Y: HPLLOT X + 2
    ,Y TO X + 2,Y - 6: HPLLOT X + 1,Y - 5
2890 RETURN
2900 REM 2
2910 HPLLOT X,Y TO X + 4,Y: HPLLOT X,Y - 1: HPLLOT
    X + 1,Y - 2
2920 HPLLOT X + 2,Y - 3 TO X + 3,Y - 3: HPLLOT
    X + 4,Y - 4 TO X + 4,Y - 5
2930 HPLLOT X + 1,Y - 6 TO X + 3,Y - 6: HPLLOT
    X,Y - 5
2940 RETURN
2950 REM 3
2960 HPLLOT X,Y - 6 TO X + 4,Y - 6: HPLLOT X
    + 1,Y TO X + 3,Y: HPLLOT X + 2,Y - 3 TO
    X + 3,Y - 3
2970 HPLLOT X,Y - 1: HPLLOT X + 4,Y - 1 TO X
    + 4,Y - 2: HPLLOT X + 3,Y - 4: HPLLOT X
    + 4,Y - 5
2980 RETURN
2990 REM 4

```

```

3000 H PLOT X + 3,Y TO X + 3,Y - 6: H PLOT X
      ,Y - 2 TO X + 4,Y - 2
3010 H PLOT X,Y - 3: H PLOT X + 1,Y - 4: H PLOT
      X + 2,Y - 5
3020 RETURN
3030 REM 5
3040 H PLOT X,Y - 6 TO X + 4,Y - 6: H PLOT X
      ,Y - 4 TO X + 3,Y - 4: H PLOT X + 4,Y -
      1 TO X + 4,Y - 3: H PLOT X + 1,Y TO X +
      3,Y
3050 H PLOT X,Y - 5: H PLOT X,Y - 1
3060 RETURN
3070 REM 6
3080 H PLOT X + 1,Y TO X + 3,Y: H PLOT X,Y -
      3 TO X + 3,Y - 3: H PLOT X + 2,Y - 6 TO
      X + 4,Y - 6
3090 H PLOT X + 1,Y - 5: H PLOT X,Y - 4
3100 H PLOT X,Y - 1 TO X,Y - 2: H PLOT X + 4
      ,Y - 1 TO X + 4,Y - 2
3110 RETURN
3120 REM 7
3130 H PLOT X + 1,Y TO X + 1,Y - 2: H PLOT X
      ,Y - 6 TO X + 4,Y - 6
3140 H PLOT X + 2,Y - 3: H PLOT X + 3,Y - 4:
      H PLOT X + 4,Y - 5
3150 RETURN
3160 REM R
3170 H PLOT X,Y TO X,Y - 6: H PLOT X + 1,Y -
      6 TO X + 3,Y - 6: H PLOT X + 1,Y - 3 TO
      X + 3,Y - 3: H PLOT X + 4,Y - 4 TO X +
      4,Y - 5
3180 H PLOT X + 2,Y - 2: H PLOT X + 3,Y - 1:
      H PLOT X + 4,Y
3190 RETURN
3200 REM 0
3210 H PLOT X,Y - 1 TO X,Y - 5: H PLOT X + 4
      ,Y - 1 TO X + 4,Y - 5: H PLOT X + 1,Y TO
      X + 3,Y: H PLOT X + 1,Y - 6 TO X + 3,Y -
      6
3220 RETURN
3230 REM W
3240 H PLOT X,Y TO X,Y - 6: H PLOT X + 4,Y TO
      X + 4,Y - 6
3250 H PLOT X + 1,Y - 1: H PLOT X + 3,Y - 1:
      H PLOT X + 2,Y - 2: H PLOT X + 2,Y - 3
3260 RETURN

```

Beans

No, this isn't a recipe for three-bean salad. It's a two-player game of logic designed to nourish your intellect. You're maneuvering small beans, trying to get as many as possible onto your platter.

Normally, "Beans" is played with small pieces and a tray with shallow depressions. You and your opponent each start with 18 beans, divided equally among the six depressions, or cups. The object is to move as many beans as you can into your Home. With the computer, of course, it's all displayed on the screen. You pick up and deposit the beans by pressing a single key. You don't even have to bother to count the beans out—the computer does that, too.

One Bean in Each Cup

You're ready to play Beans as soon as you've typed in the program, saved, and run it. The rules are a bit complicated, especially if you haven't played the game before. They appear on the screen once you've typed RUN. Since you may want to look them over carefully, they're repeated here.

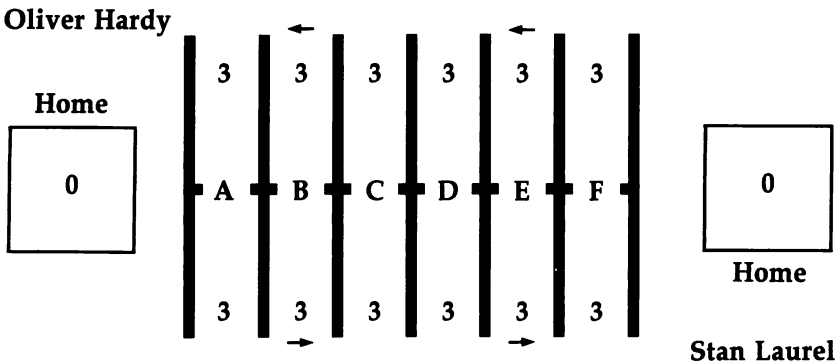
- A move is made by taking all the beans from any cup on *your* side of the board and sowing them counterclockwise, one bean at a time. In other words, one bean is deposited in each of the cups, until there are no more beans to place.
- If you sow the last bean of your move in your Home cup, you get another move.
- If the last bean sown lands in a beanless cup, with the cup opposite *not* empty, *all* the beans in the two cups are moved to your Home.
- The game ends when either player has no beans remaining in any of the six cups on that player's side of the board. The player with the most beans at Home wins.

It may be easier to understand if you look at an example. Make sure the board is displayed on the screen so that you can refer to it if you want to duplicate the example moves.

Sample Beans

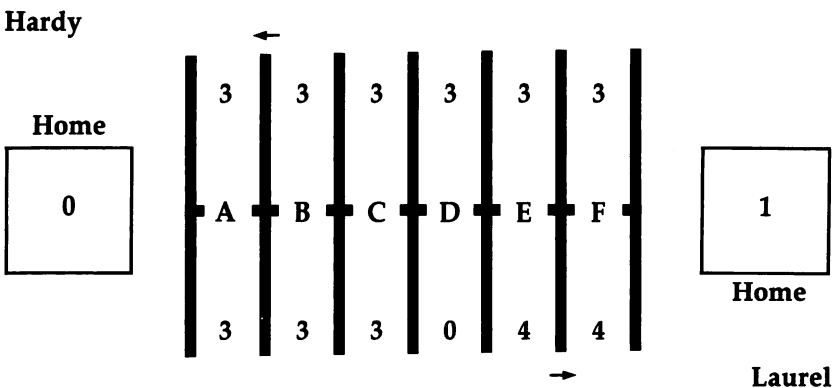
Two players, Laurel and Hardy, are about to begin a game of Beans. Take a look at Figure 1, which shows the initial board display.

Figure 1. Initial Positions



Laurel (who controls the cups at the bottom of the board and whose Home is at the right) goes first and presses the D key to tell the computer to sow the beans from cup D. The result is illustrated in Figure 2.

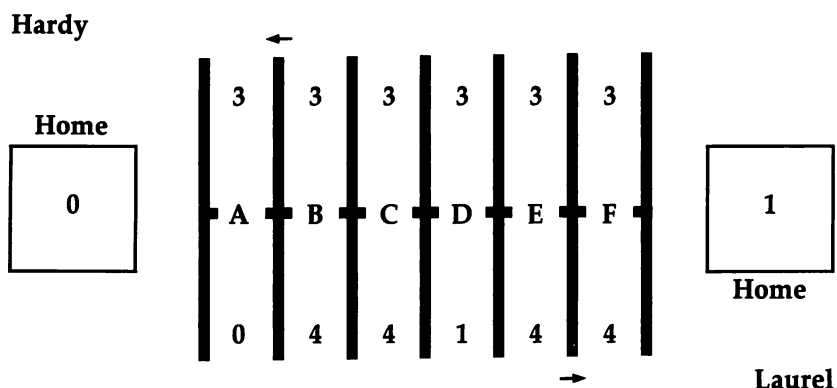
Figure 2. Laurel's Move



The three beans in cup D are picked up; the first bean is deposited in cup E, the second in cup F, and the last in Laurel's Home platter. Cup D is now empty, cups E and F now have four beans each, and Home has one. Since the last bean

was placed in Home, Laurel gets to move again. This time he elects to sow the beans from cup A. Look at Figure 3 to see what happens.

Figure 3. Laurel Moves Again



Three beans are picked up and deposited in the next three cups. Both cup B and cup C now have four beans, while cup D has one. That last bean was important. It landed in an empty cup. But the opposite cup (D, top) wasn't empty. So all the beans in both cups labeled D (a total of four) are Laurel's. When Laurel presses a key, they're automatically placed in his Home. Cup D will then show zero beans, while Home increases to five.

Now it's Hardy's turn to sow. The game continues until one player is left without any beans on his side of the board. Again, the player with the most beans at Home at the end of the game wins.

Beans

```

100 REM BEANS
110 REM INITIALIZE
120 GOSUB 230
130 REM PLAY GAME
140 GOSUB 720
150 REM OUTCOME
160 GOSUB 2470
170 REM PLAY AGAIN
180 VTAB 23: HTAB 11: PRINT "Play again (Y
  /N) ? "; CHR$ (7);
190 GET S$
200 IF S$ = "Y" OR S$ = "y" THEN 140
    
```

```

210 IF S$ < > "N" AND S$ < > "n" THEN 18
    0
220 END
230 REM INITIALIZE
240 REM TITLE
250 GOSUB 310
260 REM INSTRUCTIONS
270 GOSUB 360
280 REM PLAYERS
290 GOSUB 600
300 RETURN
310 REM TITLE
320 PRINT CHR$(21): TEXT : HOME
330 VTAB 13: HTAB 18: PRINT "Beans
340 FOR PAUSE = 1 TO 2000: NEXT
350 RETURN
360 REM INSTRUCTIONS
370 HOME
380 PRINT "You and your opponent start wit
    h 18
390 PRINT "beans each, divided into six pi
    ts.
400 PRINT
410 PRINT "A MOVE is made by taking all of
    the
420 PRINT "beans from any pit on your side
    of the
430 PRINT "board and sowing them countercl
    ockwise,"
440 PRINT "one bean at a time.
450 PRINT
460 PRINT "If you sow the last bean of you
    r MOVE
470 PRINT "in your HOME pit, you get anothe
    r MOVE."
480 PRINT
490 PRINT "If the last bean sown lands in
    a
500 PRINT "beanless pit, with the opposite
    pit
510 PRINT "not empty, all the beans in the
    two
520 PRINT "pits are yours.
530 PRINT
540 PRINT "The game ends when either side
    is
550 PRINT "stripped. The player with the m
    ost
560 PRINT "beans wins.
570 VTAB 23: HTAB 14: PRINT "Press any key
    ";

```

Brain teasers

```
580 GET S$
590 RETURN
600 REM PLAYERS
610 HOME : DIM B(14)
620 PRINT "Two players are required. Please enter
630 PRINT "your names, with up to 10 characters
640 PRINT "allowed.
650 FOR I = 1 TO 2
660 VTAB 5: HTAB 16: PRINT SPC( 25); CHR$(7)
670 VTAB 5: HTAB 1: PRINT "Name No. ";I;: INPUT
    " = ? ";N$
680 IF N$ = "" OR LEN (N$) > 10 THEN 660
690 NM$(I) = N$
700 NEXT
710 RETURN
720 REM GAME
730 REM DRAW BOARD
740 GOSUB 960
750 REM LABEL
760 GOSUB 1250
770 REM INITIAL VALUES
780 GOSUB 1500
790 REM SELECT MOVE
800 GOSUB 1590
810 REM MAKE IT
820 GOSUB 1900
830 REM CHECK FOR EXTRA POINTS
840 IF B(LAST) = 1 AND LAST < > 7 AND LAST < > 14 THEN GOSUB 2140
850 REM CHECK FOR GAME OVER
860 GOSUB 2350
870 IF GAME$ = "OVER" THEN 950
880 REM CHECK FOR ANOTHER MOVE BY SAME PLAYER
890 GOSUB 2430
900 IF M$ = "MOVE AGAIN" THEN 800
910 REM TAKE TURNS
920 PLAYER = PLAYER + 1
930 IF PLAYER = 3 THEN PLAYER = 1
940 IF GAME$ = "ON" THEN 800
950 RETURN
960 REM DRAW BOARD
970 REM PITS
980 HOME
990 GOSUB 1030
1000 REM HOMES
1010 GOSUB 1130
```

```

1020 RETURN
1030 REM PITS
1040 HOME : INVERSE
1050 FOR X = 8 TO 32 STEP 4
1060 FOR Y = 7 TO 15
1070 VTAB Y: HTAB X: PRINT CHR$ (32)
1080 NEXT Y,X
1090 FOR X = 8 TO 32
1100 VTAB 11: HTAB X: PRINT CHR$ (32)
1110 NEXT
1120 RETURN
1130 REM HOMES
1140 REM TOPS & BOTTOMS
1150 FOR I = 1 TO 6
1160 VTAB 9: HTAB I: PRINT CHR$ (32): VTAB
13: HTAB I: PRINT CHR$ (32)
1170 VTAB 9: HTAB 33 + I: PRINT CHR$ (32)
: VTAB 13: HTAB 33 + I: PRINT CHR$ (3
2)
1180 NEXT
1190 REM SIDES
1200 FOR I = 1 TO 3
1210 VTAB 9 + I: HTAB 1: PRINT CHR$ (32):
VTAB 9 + I: HTAB 6: PRINT CHR$ (32)
1220 VTAB 9 + I: HTAB 34: PRINT CHR$ (32)
: VTAB 9 + I: HTAB 39: PRINT CHR$ (32
)
1230 NEXT
1240 RETURN
1250 REM LABEL
1260 VTAB 1: HTAB 13: PRINT "THE BEAN BOAR
D"
1270 VTAB 7: HTAB 2: PRINT "HOME": VTAB 15
: HTAB 35: PRINT "HOME": NORMAL
1280 REM LETTERS
1290 X = 10
1300 FOR I = 65 TO 70
1310 VTAB 11: HTAB X: PRINT CHR$ (I)
1320 X = X + 4
1330 NEXT
1340 REM NUMBERS
1350 X = 10
1360 FOR I = 1 TO 6
1370 VTAB 8: HTAB X: PRINT "3"
1380 VTAB 14: HTAB X: PRINT "3"
1390 X = X + 4
1400 NEXT
1410 VTAB 11: HTAB 3: PRINT "0": VTAB 11: HTAB
36: PRINT "0"
1420 REM ARROWS

```

```

1430 INVERSE
1440 VTAB 17: HTAB 12: PRINT "---->"
1450 VTAB 5: HTAB 25: PRINT "<----": NORMAL

1460 REM NAMES
1470 VTAB 5: HTAB 2: PRINT NM$(2)
1480 VTAB 17: HTAB 39 - LEN (NM$(1)): PRINT
    NM$(1)
1490 RETURN
1500 REM VALUES
1510 FOR I = 1 TO 6
1520 B(I) = 3
1530 B(I + 7) = 3
1540 NEXT
1550 B(7) = 0: B(14) = 0
1560 GAME$ = "ON"
1570 PLAYER = 1
1580 RETURN
1590 REM SELECT MOVE
1600 REM CLEAR LINES
1610 GOSUB 1790
1620 VTAB 19: HTAB 1
1630 IF M$ = "" THEN PRINT "Your turn to
    sow ";
1640 IF M$ = "MOVE AGAIN" THEN PRINT "You
    get to sow again ";
1650 PRINT NM$(PLAYER); "."
1660 VTAB 21: HTAB 1: PRINT "Pit to take f
    rom = ? "
1670 VTAB 21: HTAB 22: PRINT SPC( 1); CHR$
    (7);: HTAB 22
1680 GET S$
1690 A = ASC (S$)
1700 IF A > 90 THEN A = A - 32
1710 IF A < 65 OR A > 70 THEN 1670
1720 PRINT CHR$ (A)
1730 REM BIN NUMBER
1740 N = 78 - A
1750 IF PLAYER = 1 THEN N = A - 64
1760 REM CHECK FOR EMPTY BIN
1770 IF B(N) = 0 THEN GOSUB 1850: GOTO 16
    70
1780 RETURN
1790 REM CLEAR BOTTOM LINES
1800 VTAB 23: HTAB 1: PRINT SPC( 39)
1810 FOR I = 19 TO 21
1820 VTAB I: HTAB 1: PRINT SPC( 39)
1830 NEXT
1840 RETURN
1850 REM EMPTY BIN

```

```

1860 VTAB 23: HTAB 15: FLASH : PRINT "EMPTY
    Y PIT !": NORMAL
1870 FOR PAUSE = 1 TO 250: SOUND = PEEK ( -
    16336): NEXT
1880 VTAB 23: HTAB 15: PRINT SPC( 11)
1890 RETURN
1900 REM MAKE MOVE
1910 K = B(N)
1920 REM REMOVE ALL BEANS FROM ORIGINAL P
    IT
1930 PIT = N
1940 B(N) = 0
1950 GOSUB 2060
1960 REM ADD BEANS TO OTHER PITS
1970 FOR I = 1 TO K
1980 PIT = PIT + 1
1990 IF PIT = 15 THEN PIT = 1
2000 B(PIT) = B(PIT) + 1
2010 REM DISPLAY NEW VALUE
2020 GOSUB 2060
2030 NEXT
2040 LAST = PIT
2050 RETURN
2060 REM DISPLAY NEW VALUE
2070 IF PIT = 7 THEN R = 11: C = 36
2080 IF PIT = 14 THEN R = 11: C = 3
2090 IF PIT > = 1 AND PIT < = 6 THEN R =
    14: C = 6 + 4 * PIT
2100 IF PIT > = 8 AND PIT < = 13 THEN R =
    8: C = 62 - 4 * PIT
2110 VTAB R: HTAB C: PRINT SPC( 2)
2120 VTAB R: HTAB C: PRINT B(PIT)
2130 RETURN
2140 REM CHECK FOR EXTRA POINTS
2150 REM NUMBER OF THE OPPOSITE BIN
2160 NOPP = 14 - PIT
2170 IF B(NOPP) = 0 THEN 2340
2180 REM DISPLAY MESSAGE
2190 L$ = CHR$ (78 - LAST)
2200 IF LAST < = 6 THEN L$ = CHR$ (LAST +
    64)
2210 VTAB 19: HTAB 1: PRINT SPC( 39)
2220 VTAB 19: HTAB 1: PRINT "Congratulatio
    ns "; NM$(PLAYER); " ! Your last"
2230 PRINT "bean was sown in an empty pit,
    with
2240 PRINT "the opposite pit not empty.
2250 VTAB 23: HTAB 1: PRINT "Hit any key t
    o reap the beans in "; L$; " "; CHR$ (7)
    ;

```

```

2260 GET S$
2270 REM ADD BEANS TO HOME
2280 IF PLAYER = 1 THEN B(7) = B(7) + B(LA
ST) + B(NOPP):PIT = 7
2290 IF PLAYER = 2 THEN B(14) = B(14) + B(
LAST) + B(NOPP):PIT = 14
2300 GOSUB 2060
2310 REM SUBTRACT BEANS
2320 B(LAST) = 0:PIT = LAST: GOSUB 2060
2330 B(NOPP) = 0:PIT = NOPP: GOSUB 2060
2340 RETURN
2350 REM CHECK FOR GAME OVER
2360 T1 = 0:T2 = 0
2370 FOR I = 1 TO 6
2380 T1 = T1 + B(I)
2390 T2 = T2 + B(I + 7)
2400 NEXT
2410 IF T1 = 0 OR T2 = 0 THEN GAME$ = "OVE
R"
2420 RETURN
2430 REM CHECK FOR ANOTHER MOVE
2440 M$ = ""
2450 IF (PLAYER = 1 AND LAST = 7) OR (PLAY
ER = 2 AND LAST = 14) THEN M$ = "MOVE
AGAIN"
2460 RETURN
2470 REM OUTCOME
2480 REM CLEAR LINES
2490 GOSUB 1790
2500 VTAB 23: HTAB 13: FLASH : PRINT "THE
GAME'S OVER !": NORMAL
2510 FOR PAUSE = 1 TO 30: PRINT CHR$ (7);
: NEXT
2520 VTAB 23: HTAB 13: PRINT SPC( 17)
2530 VTAB 19: HTAB 1
2540 IF B(7) = B(14) THEN PRINT "At least
neither of you lost !"
2550 IF B(7) > B(14) THEN PRINT "Congratu
lations ";NM$(1);" !"
2560 IF B(14) > B(7) THEN PRINT "Congratu
lations ";NM$(2);" !"
2570 RETURN

```


2

Personal Planners

1

2

3

4

5

6

7

8

9

10

Personal Planners

Computers have been applied with much success in the realms of personal financial and household management. Computations which are tedious to perform even with a standard calculator are easy to manage with a simple BASIC program.

This chapter presents a series of programs designed to help you better manage your personal finances. The programs are easy to use and apply. Also included are two other programs which are handy to have around the home.

These programs do not provide a miracle cure for your financial ills. They won't turn you into a stock market forecaster or a financial advisor. But they *will* help you manage your own finances.

Because the programs are printed here, not just on a disk like commercial software, you can see how they're put together. And, if you're so inclined, you can even modify them.

Before you begin entering these programs, make sure you read the Appendix, "Beginner's Guide to Typing In Programs." All the programs are printed as you should type them in.

The 12 home application programs in this chapter are:

- **IRA Planner.** Calculate what your Individual Retirement Account (IRA) will be worth at maturity, in both today's dollars and in dollars adjusted for inflation.
- **Car Payments.** Compute the monthly payment on a car loan, as well as the total payment made over the life of the loan.
- **Mortgage Payments.** Figure yearly, quarterly, or monthly mortgage payments; the total payment over the life of the loan; and the amount paid to principal and to interest.
- **College Planner.** Decide how much money you'll need to save to provide higher education for your child.
- **Effective Yield on Investments.** Calculate the actual return on your money, using several frequencies of interest compounding.
- **Municipal Bond Buyer.** Figure the cost of municipal bonds, their yields (both tax-free and taxable-equivalent), and their future worth.

- **Household Budget.** Develop a plan for balancing your budget or for reaching a savings goal.
- **Car Trek.** Compute your car's gas mileage.
- **Weather Forecaster.** Learn how to make accurate, short-range weather forecasts.
- **Calorie Cop.** See how much energy you'll burn up in performing a host of activities.
- **Utility-Bill Audit.** Check your bills for accuracy, or use it to split costs among the people living in your household.
- **Paycheck Analysis.** Verify the accuracy of your take-home pay, or even calculate it for your employees.

You'll find all these programs worth the short time it takes to type them in. They'll provide an instant library of financial aids and planning tools. You won't have to spend time and money getting information that is simple and easy to obtain. The programs don't tell you what to do; that's still up to you. But they *will* help you make those tough financial decisions for yourself.

IRA Planner

Individual Retirement Accounts (IRAs) are being generated in record numbers. This program will help you plan your IRA contributions and tell you how much you'll have saved by retirement.

IRAs are becoming increasingly popular. Not only are they a way to save for your retirement, but funds in IRAs are not taxed until they're withdrawn (usually when you turn 59½). Also, you can deduct the money placed in an IRA from your federal tax liability. You actually reap double benefits from IRAs—you get a tax deduction now and savings for later. If you're in a high tax bracket, IRAs are especially attractive. Since you assume you'll be making less money when you retire, you'll be taxed at a lower rate when you *do* withdraw the money. For many people, IRAs are a good idea.

There is a catch, however. You can't withdraw money from your IRAs before age 59½ without suffering a stiff penalty. You must also begin to withdraw the money before age 70½. And though IRAs may make many of us millionaires in 35 or 40 years if interest rates stay high, inflation may mercilessly erode the buying power of those future dollars.

Interest and Inflation

Type in and save "IRA Planner." Run the program and you'll see the first of a series of display screens. Each asks for a figure—when you want to begin withdrawing the money, your current age, the size of your annual deposit, the projected interest rate, and the expected rate of inflation.

Enter the numbers as requested. The interest and inflation rates will be only estimations on your part, of course. There's no way you can gaze into the future. If you could, you probably wouldn't need an IRA. The best thing is to plot your IRA contributions several times, using different interest and inflation rates. That way you can get a better idea of what will happen to your money under several different conditions. Once you've entered the estimated rate of inflation, the program calculates your IRA value at maturity and presents a report.

You'll see several things on the screen. Your age, the span of your IRA, the expected rates of interest and inflation, your annual payments, and your total payments are all self-explanatory. The important figures are near the bottom of the screen.

Current dollars. This is the amount you'd actually see listed in your IRA account after the number of years you specified for contributing have passed.

Constant dollars. This is the amount of money you'll end up with, *adjusted for inflation*. In other words, this is how much your IRA will be worth in terms of today's dollar value. Sometimes, if you've estimated that the interest and inflation rates will be nearly the same, the figure here will be *less than* the amount you put into the IRA. For instance, if you put \$60,000 into an IRA over 30 years, and entered 10 and 9 percent for interest and inflation rates, respectively, the amount here will be only \$27,275. That's less than the money you put into the account. (Remember, this figure is in *present* dollars.) Because of inflation, the contributions in the later years of your account are actually worth less each year. You're not really putting the same in each year, since as inflation's effect is calculated, your contributions decrease *in respect to present dollars*.

A Millionaire?

Let's go through a quick example. Run the program and enter 2 at the first prompt. This instructs the program to calculate the IRA as if you were going to wait until you're 70½ (it's rounded off to 70 in the program) before taking money out. As an example, type in 33 for your age and 2000 for your annual contribution. The program automatically places dollar signs, commas, and decimal points. Then enter 12 and 5 for the expected interest and inflation rates, respectively.

Your hypothetical IRA will be worth \$1.2 million when you're 70. In today's dollars, this sum is worth just \$200,000. Your total IRA payments over the 37-year span of the account equal only \$74,000. Half of that may have gone to the IRS anyway, even if you hadn't invested in an IRA (we'll assume a 50 percent tax bracket). Keeping this in mind, the constant dollar payoff is actually pretty high.

Run the program as many times as you like, inserting new figures each time. This will give you an idea of what will hap-

pen to your IRA under a wide variety of conditions. Remember, however, that the program doesn't tell you to invest in an IRA, or even when. Those decisions are up to you.

IRA Planner

```

100 REM  IRA PLANNER
110 REM  HEADING
120 GOSUB 200
130 REM  ENTER DATA
140 GOSUB 480
150 REM  COMPUTE
160 GOSUB 1000
170 REM  DISPLAY RESULTS
180 GOSUB 1130
190 END
200 REM  HEADING
210 PRINT CHR$(21): TEXT : HOME
220 REM  TITLE
230 GOSUB 270
240 REM  EXPLANATION
250 GOSUB 360
260 RETURN
270 REM  TITLE
280 HOME
290 VTAB 12: HTAB 15: PRINT "Individual
300 PRINT TAB(16)"Retirement
310 PRINT TAB(17)"Account
320 PRINT TAB(18)"Planner
330 FOR PAUSE = 1 TO 2500: NEXT
340 L$ = "": FOR I = 1 TO 39: L$ = L$ + "=":
    NEXT
350 RETURN
360 REM  EXPLANATION
370 HOME
380 PRINT "This program computes the value
    of your
390 PRINT "IRA at age 59 and 70.
400 PRINT
410 PRINT "59 is the earliest age that you
    can
420 PRINT "start withdrawing funds without
    penalty.
430 VTAB 7: HTAB 1: PRINT "And 70 is the l
    atest age that you can
440 PRINT "delay withdrawing.
450 VTAB 23: HTAB 14: PRINT "Press any key
    ";
460 GET S$
470 RETURN

```

```

480 REM ENTER DATA
490 REM SELECT 59 OR 70
500 GOSUB 600
510 REM AGE
520 GOSUB 710
530 REM YEARLY DEPOSIT
540 GOSUB 790
550 REM INTEREST RATE
560 T$ = "Interest": GOSUB 870:RI = RT
570 REM INFLATION RATE
580 T$ = "Inflation": GOSUB 870:RF = RT
590 RETURN
600 REM SELECT 59 OR 70
610 HOME
620 PRINT "Would you like IRA computations
made
630 PRINT "through
640 VTAB 4: HTAB 10: INVERSE : PRINT "1";:
NORMAL : PRINT " Age 59, or
650 VTAB 6: HTAB 10: INVERSE : PRINT "2";:
NORMAL : PRINT " Age 70
660 VTAB 8: HTAB 10: PRINT "Choice (1 or 2
) = ? "; CHR$ (7);
670 GET S$
680 CHOICE = VAL (S$)
690 IF CHOICE < 1 OR CHOICE > 2 THEN 660
700 RETURN
710 REM AGE
720 VTAB 12: HTAB 1: PRINT "Pardon my aski
ng, but how old are
730 VTAB 13: HTAB 7: PRINT SPC( 15); CHR$
(7)
740 VTAB 13: HTAB 1: INPUT "you ? ";A$
750 AGE = VAL (A$)
760 IF AGE < = 0 OR AGE > 125 THEN VTAB
23: HTAB 10: PRINT "Who are you kiddin
g !": GOTO 730
770 IF (CHOICE = 1 AND AGE > = 59) OR (CH
OICE = 2 AND AGE > = 70) THEN VTAB 2
3: HTAB 1: PRINT "Sorry, an IRA probab
ly isn't for you.": STOP
780 RETURN
790 REM YEARLY DEPOSIT
800 HOME
810 PRINT "How much do you want to put int
o your
820 VTAB 2: HTAB 20: PRINT SPC( 15); CHR$
(7)
830 VTAB 2: HTAB 1: INPUT "IRA each year ?
";D$

```



```

840 DPT = VAL (D$)
850 IF DPT < = 0 THEN 820
860 RETURN
870 REM RATES
880 HOME
890 PRINT "Please enter the ";T$;" Rate th
    at
900 PRINT "you expect will prevail, on ave
    rage,
910 PRINT "over the life of your IRA
920 PRINT
930 PRINT "For example, enter 7 for 7%, 10
    for
940 PRINT "10%, and so on.
950 VTAB 8: HTAB 19: PRINT SPC( 15); CHR$
    (7)
960 VTAB 8: HTAB 1: PRINT T$;" Rate";: INPUT
    " = ? ";R$
970 RT = VAL (R$)
980 IF RT < = 0 THEN 950
990 RETURN
1000 REM COMPUTE
1010 Y = 59
1020 IF CHOICE = 2 THEN Y = 70
1030 REM YEARS UNTIL MATURITY
1040 N = Y - AGE
1050 REM CURRENT-DOLLAR VALUE
1060 FOR I = 1 TO N
1070 VL = VL + DPT * (1 + RI / 100) ^ (N -
    I + 1)
1080 NEXT
1090 REM DEFLATE
1100 DF = (1 + RF / 100) ^ N
1110 KVL = VL / DF
1120 RETURN
1130 REM RESULTS
1140 REM HEADING
1150 GOSUB 1220
1160 REM BODY
1170 GOSUB 1280
1180 VTAB 22: HTAB 1: PRINT L$
1190 VTAB 23: HTAB 14: PRINT "Press any ke
    y ";
1200 GET S$
1210 RETURN
1220 REM HEADING
1230 HOME
1240 PRINT L$
1250 PRINT TAB( 6)"I.R.A. PAYOFF THROUGH
    AGE ";Y

```

```

1260 PRINT L$
1270 RETURN
1280 REM BODY
1290 VTAB 5: HTAB 1: PRINT "Your Age"; TAB(
18)"=";:HT = 30:NR = AGE:DA = 0: GOSUB
9000
1300 VTAB 5: HTAB 32: PRINT "Years
1310 VTAB 6: HTAB 1: PRINT "IRA Span"; TAB
(18)"=";:NR = N: GOSUB 9000
1320 VTAB 6: HTAB 32: PRINT "Years
1330 VTAB 8: HTAB 1: PRINT "Expected
1340 PRINT "Interest Rate"; TAB( 18)"=";:H
T = 33:NR = RI:DA = 2: GOSUB 9000
1350 VTAB 9: HTAB 35: PRINT "%"
1360 VTAB 11: HTAB 1: PRINT "Expected
1370 PRINT "Inflation Rate"; TAB( 18)"=";:
NR = RF: GOSUB 9000
1380 VTAB 12: HTAB 35: PRINT "%"
1390 VTAB 14: HTAB 1: INVERSE : PRINT "PAY
MENTS": NORMAL
1400 DOLL$ = "YES"
1410 VTAB 15: HTAB 1: PRINT "Annual"; TAB(
18)"=";:NR = DPT: GOSUB 9000
1420 VTAB 16: HTAB 1: PRINT "Total"; TAB(
18)"=";:NR = N * DPT: GOSUB 9000
1430 VTAB 18: HTAB 1: INVERSE : PRINT "IRA
PAYOFF": NORMAL
1440 VTAB 19: HTAB 1: PRINT "Current Dolla
rs"; TAB( 18)"=";:NR = VL: GOSUB 9000
1450 VTAB 20: HTAB 1: PRINT "Constant Doll
ars"; TAB( 18)"=";:NR = KVL: GOSUB 900
0
1460 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Car Payments

If you are planning to borrow money to purchase a car, you'll find this short program helpful in estimating your payments.

After many months, you've finally succumbed to your dream of owning that new car. You haggle with the dealer and finally get the price within reason. But you wonder if you can afford the payments.

It's easy to have that information *before* you walk into the dealership, bank, or credit union. All you have to do is run this short program on your computer. By inserting the amount of your loan, its length, and the interest rate, you can quickly see how much your monthly payments will be, as well as the total amount of interest you'll pay over the life of the loan.

Loan Officer

You can have this information at your fingertips, just as your loan officer does, by typing in and saving "Car Payments." Run it, and you'll cycle through a number of screen displays, each of which asks for a different piece of data.

Amount borrowed. You can enter any amount, up to \$999,999. (Not too many cars are worth that much.) Enter whole numbers, not fractions or decimals. The program automatically places dollar signs, commas, and decimal points.

Length of loan. Type in the length of the loan, in years and months. Three years, for instance, can be entered as 3 years, 0 months.

Interest rate. Enter the interest rate you expect to pay for your car loan. You'll probably have to call your bank or credit union, or the dealership, to find out some possible interest rates. You can enter decimals in this category. For example, 12-1/2 percent interest would be typed in as 12.5.

The program takes only a moment to compute your loan summary. It summarizes your entries and then tells you the amount of your monthly payment, the total you'll pay, the principal (which should be identical to the amount borrowed), and the total interest to be paid.

Pressing any key at the end stops the program. If you want to go through it again, to see the results of a different interest rate, for example, type RUN and the program starts over.

Buy a New Car

As an example, we'll borrow \$5,000 for three years. Assume we'll get a great deal on financing the loan, so our interest rate will be 12.5 percent. After entering those numbers, the computer pauses a moment and then displays the loan summary.

The monthly payments will be \$167.27. The total amount you'll pay is \$6,021.65, with \$1,021.65 being interest on the loan. If you want to know what the payments would be if you stretched out the loan to four years, all you have to do is run the program again, using the new data. It's that easy.

Car Payments

```

100 REM CAR PAYMENTS
110 REM ENTER DATA
120 GOSUB 180
130 REM COMPUTE
140 GOSUB 620
150 REM DISPLAY RESULTS
160 GOSUB 740
170 END
180 REM ENTER DATA
190 PRINT CHR$(21): TEXT : HOME
200 L$ = "": FOR I = 1 TO 39:L$ = L$ + " ":
    NEXT
210 PRINT L$
220 PRINT TAB(14)"CAR PAYMENTS
230 PRINT L$
240 REM AMOUNT OF LOAN
250 GOSUB 310
260 REM PERIOD OF LOAN
270 GOSUB 400
280 REM INTEREST RATE
290 GOSUB 530
300 RETURN
310 REM LOAN
320 VTAB 5: HTAB 1: PRINT "How much money
    would you like to
330 VTAB 6: HTAB 10: PRINT SPC(15); CHR$(
    7)
340 VTAB 6: HTAB 1: INPUT "borrow ? ";B$
350 LOAN = VAL (B$)
360 VTAB 23: HTAB 7: PRINT SPC(33)
370 IF LOAN < = 0 THEN VTAB 23: HTAB 8: PRINT
    "Please borrow something !": GOTO 330
380 IF LOAN > 1E6 THEN VTAB 23: HTAB 7: PRINT
    "Please scale down figure !": GOTO 330
390 RETURN
400 REM PERIOD OF LOAN

```

```

410 VTAB 9: HTAB 1: PRINT "Please enter the
    length of your loan in
420 PRINT "years and months.
430 VTAB 12: HTAB 12: PRINT SPC( 15); CHR$
    (7)
440 VTAB 12: HTAB 2: INPUT "Years = ? ";Y$
450 Y = INT ( VAL (Y$))
460 IF Y < 0 THEN 430
470 VTAB 13: HTAB 12: PRINT SPC( 15); CHR$
    (7)
480 VTAB 13: HTAB 1: INPUT "Months = ? ";M
    $
490 M = INT ( VAL (M$))
500 IF M < 0 OR M > 12 THEN 470
510 N = Y * 12 + M
520 RETURN
530 REM INTEREST RATE
540 VTAB 16: HTAB 1: PRINT "Please enter the
    interest rate on your
550 PRINT "loan. For example, enter 8 for
    8%,
560 PRINT "11 for 11%, and so on.
570 VTAB 20: HTAB 19: PRINT SPC( 15); CHR$
    (7)
580 VTAB 20: HTAB 1: INPUT "Interest Rate
    = ? ";R$
590 RT = VAL (R$)
600 IF RT < = 0 THEN VTAB 23: HTAB 8: PRINT
    "There's no free lunch !": GOTO 570
610 RETURN
620 REM COMPUTE
630 REM INTEREST RATE PER PERIOD
640 R = (RT / 100) / 12
650 REM PAYMENT PER PERIOD
660 REM NUMERATOR
670 P1 = LOAN * R * (1 + R) ^ N
680 REM DENOMINATOR
690 P2 = (1 + R) ^ N - 1
700 PPP = P1 / P2
710 REM TOTAL PAYMENT
720 TPAYMENT = N * PPP
730 RETURN
740 REM DISPLAY
750 HOME
760 PRINT L$
770 PRINT TAB( 10)"SUMMARY OF THE LOAN
780 PRINT L$
790 VTAB 5: HTAB 1: INVERSE : PRINT "LOAN
    VALUES": NORMAL

```

```

800  VTAB 7: HTAB 1: PRINT "Amount"; TAB( 1
      8) "="; DA = 2: DOLL$ = "YES"; NR = LOAN;
      HT = 33: GOSUB 9000
810  VTAB 8: HTAB 1: PRINT "Number of years
      "; TAB( 18) "="; DA = 0: DOLL$ = ""; HT =
      30: NR = Y: GOSUB 9000
820  VTAB 9: HTAB 1: PRINT "Number of month
      s"; TAB( 18) "="; NR = M: GOSUB 9000
830  VTAB 10: HTAB 1: PRINT "Interest Rate"
      ; TAB( 18) "="; DA = 2: HT = 33: NR = RT:
      GOSUB 9000
840  VTAB 10: HTAB 35: PRINT "%"
850  VTAB 14: HTAB 1: INVERSE : PRINT "LOAN
      PAYMENTS"; NORMAL
860  VTAB 16: HTAB 1: PRINT "Monthly"; TAB(
      18) "="; DOLL$ = "YES"; NR = PPP: GOSUB
      9000
870  VTAB 18: HTAB 1: PRINT "Total"; TAB( 1
      8) "="; NR = TPAYMENT: GOSUB 9000
880  VTAB 19: HTAB 3: PRINT "Principal"; TAB(
      18) "="; NR = LOAN: GOSUB 9000
890  VTAB 20: HTAB 3: PRINT "Interest"; TAB(
      18) "="; NR = TPAYMENT - LOAN: GOSUB 90
      00
900  VTAB 22: HTAB 1: PRINT L$
910  VTAB 23: HTAB 14: PRINT "Press any key
      ";
920  GET S$
930  RETURN
9000  REM "PRINT USING" SUBROUTINE
9010  ZR$ = "": ZS$ = "": ZD$ = "": ZZ$ = "0000
      00000"
9020  N$ = STR$ (NR)
9030  IF DOLL$ = "YES" THEN ZD$ = "$"
9040  IF NR < 0 THEN ZS$ = "-"
9050  LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
      1)
9060  LET ZL = INT (ZN): ZR = INT ((ZN - Z
      L) * 10 ^ DA)
9070  IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
      Z$ + STR$ (ZR), DA)
9080  ZT$ = " " + STR$ (ZL)
9090  IF MID$ (ZT$, LEN (ZT$) - 3, 1) < >
      "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
      ZR$
9100  HTAB (HT + 1 - LEN (N$)): PRINT N$
9110  RETURN

```

Mortgage Payments

Calculating annual, quarterly, or monthly payments on your mortgage (or any other amortized loan) is simple with this program. You can even see a payment-by-payment breakdown of the interest and principal paid during the course of the loan.

Buying a house is difficult enough. It's hard to know just what you can afford. With this short program, you can see what payments will be on different amounts of money borrowed. It won't tell you what house to buy, or even how much to borrow, but it will give you a better idea of what it will cost on a monthly basis.

The program is simple to use; all you have to do is input figures for the loan amount, its length, and the interest rate. Then you'll see how much your payments would be. It tells you how much interest you'll pay over the life of the loan and even outlines the amount of interest and principal in each payment.

Before You Go to the Bank

Type in and save "Mortgage Payments," and then run it. As in the previous program, "Car Payments," you'll see a number of screen displays.

Each display asks for one piece of information. Enter the amount you want to borrow, in whole numbers, without decimals or fractions. Next, enter the number of years you'll take to repay the loan. (Typical home mortgages are for 20 or 30 years.) Indicate whether you want the payments shown as annually, quarterly, or monthly. Finally, type in the interest rate of your loan. You can use decimals with this (for instance, 13.8 to indicate 13.8 percent).

The computer displays a summary of your loan almost instantly. Not only does it summarize the information you give it (as a check to make sure you fed it the right data), but it also shows totals of the amount you'll pay for your house, the principal (which should be the same as the amount borrowed), and the interest. Note the total interest paid: it's usually much larger than the amount borrowed. Paying over \$100,000 in interest on a \$50,000 loan is not unusual.

Payment by Payment

Pressing any key displays the next screen. At the top of the screen you'll see the payment total (whether it's monthly, quarterly, or annually). Then, that payment is separated into its principal and interest components. If you chose monthly payments, for example, it would show the first 12 months of the loan. Notice that the principal paid is a very small part of the total payment. That will change over the course of the loan.

Hit a key to move on to the next screen. You can look through the entire span of the loan this way. It's important that the program notes the breakdown of principal versus interest, for the latter can be used as a deduction in calculating your federal income taxes.

To illustrate how the program works, suppose you're borrowing \$50,000 at 12.5 percent interest, with payments paid monthly over a 30-year period. After you enter this data into the computer, it tells you that your constant monthly payment is roughly \$534. Total payments over the life of the loan are \$192,106, with \$50,000 paid to principal and \$142,106 paid to interest. You then see how much of *each* monthly payment goes to principal and interest, as the table shows.

Division of Monthly Payments

Year:Month	Paid to Principal	Paid to Interest
1:1	\$ 12.80	\$520.83
1:2	12.93	520.70
1:3	13.06	520.57
.	.	.
.	.	.
30:12	\$528.13	\$ 5.50

Mortgage Payments

```

100 REM MORTGAGE PAYMENTS
110 REM ENTER DATA
120 GOSUB 180
130 REM COMPUTE
140 GOSUB 700

```



```

150 REM DISPLAY RESULTS
160 GOSUB 880
170 END
180 REM ENTER DATA
190 PRINT CHR$(21); TEXT : HOME
200 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=";
NEXT
210 PRINT L$
220 PRINT TAB( 11)"MORTGAGE PAYMENTS
230 PRINT L$
240 REM ENTER AMOUNT OF LOAN
250 GOSUB 330
260 REM ENTER NUMBER OF YEARS
270 GOSUB 420
280 REM ENTER NUMBER OF PAYMENTS PER YEAR
290 GOSUB 500
300 REM ENTER INTEREST RATE
310 GOSUB 610
320 RETURN
330 REM LOAN
340 VTAB 5: HTAB 1: PRINT "How much money
would you like to
350 VTAB 6: HTAB 10: PRINT SPC( 15); CHR$(
7)
360 VTAB 6: HTAB 1: INPUT "borrow ? ";B$
370 LOAN = VAL (B$)
380 VTAB 23: HTAB 7: PRINT SPC( 33)
390 IF LOAN < = 0 THEN VTAB 23: HTAB 8: PRINT
"Please borrow something !": GOTO 350
400 IF LOAN > 1E6 THEN VTAB 23: HTAB 7: PRINT
"Please scale down figure !": GOTO 350
410 RETURN
420 REM NUMBER OF YEARS
430 VTAB 9: HTAB 35: PRINT SPC( 15); CHR$(
7)
440 VTAB 9: HTAB 1: INPUT "How many years
is your loan for ? ";Y$
450 Y = VAL (Y$)
460 VTAB 23: HTAB 8: PRINT SPC( 32)
470 IF Y < = 0 THEN VTAB 23: HTAB 10: PRINT
"That's a short loan !": GOTO 430
480 IF Y > 150 THEN VTAB 23: HTAB 8: PRINT
"You won't live that long !": GOTO 430
490 RETURN
500 REM PAYMENTS PER YEAR
510 HOME
520 PRINT "Payments can be made every:
530 VTAB 3: HTAB 3: INVERSE : PRINT "1";: NORMA
L : PRINT " Year

```

```

540 VTAB 5: HTAB 3: INVERSE : PRINT "2";: NORMA
    L : PRINT " Quarter,
550 VTAB 7: HTAB 3: INVERSE : PRINT "3";: NORMA
    L : PRINT " Month.
560 VTAB 9: HTAB 1: PRINT "Which would you
    like ? "; CHR$ (7);
570 GET S$
580 CHOICE = VAL (S$)
590 IF CHOICE < 1 OR CHOICE > 3 THEN 560
600 RETURN
610 REM INTEREST RATE
620 VTAB 12: HTAB 1: PRINT "Please enter t
    he interest rate on your
630 PRINT "loan. For example, enter 8 for
    8%,
640 PRINT "11 for 11%, and so on.
650 VTAB 16: HTAB 19: PRINT SPC( 15); CHR$
    (7)
660 VTAB 16: HTAB 1: INPUT "Interest Rate
    = ? ";R$
670 RT = VAL (R$)
680 IF RT < = 0 THEN VTAB 23: HTAB 8: PRINT
    "There's no free lunch !"; GOTO 650
690 RETURN
700 REM COMPUTE
710 REM NUMBER OF PAYMENTS PER YEAR
720 IF CHOICE = 1 THEN FQ = 1
730 IF CHOICE = 2 THEN FQ = 4
740 IF CHOICE = 3 THEN FQ = 12
750 REM TOTAL NUMBER OF PAYMENTS
760 N = Y * FQ
770 REM INTEREST RATE PER PERIOD
780 R = (RT / 100) / FQ
790 REM PAYMENT PER PERIOD
800 REM NUMERATOR
810 P1 = LOAN * R * (1 + R) ^ N
820 REM DENOMINATOR
830 P2 = (1 + R) ^ N - 1
840 PPP = P1 / P2
850 REM TOTAL PAYMENT
860 TPAYMENT = N * PPP
870 RETURN
880 REM DISPLAY RESULTS
890 REM SUMMARY
900 GOSUB 940
910 REM PAYMENTS PER PERIOD
920 GOSUB 1130
930 RETURN
940 REM SUMMARY
950 HOME

```

```

960 PRINT L$
970 PRINT TAB( 11)"SUMMARY OF THE LOAN
980 PRINT L$
990 VTAB 5: HTAB 1: INVERSE : PRINT "LOAN
    VALUES": NORMAL
1000 VTAB 7: HTAB 1: PRINT "Amount"; TAB(
    19)"=";:DA = 2:DOLL$ = "YES":NR = LOAN
    :HT = 34: GOSUB 9000
1010 VTAB 8: HTAB 1: PRINT "Number of year
    s"; TAB( 19)"=";:DOLL$ = "":NR = Y: GOSUB
    9000
1020 VTAB 9: HTAB 1: PRINT "Payments per y
    ear"; TAB( 19)"=";:NR = FQ: GOSUB 9000

1030 VTAB 10: HTAB 1: PRINT "Interest Rate
    "; TAB( 19)"=";:NR = RT: GOSUB 9000
1040 VTAB 10: HTAB 36: PRINT "%"
1050 VTAB 14: HTAB 1: INVERSE : PRINT "LOA
    N PAYMENTS": NORMAL
1060 VTAB 16: HTAB 1: PRINT "Total"; TAB(
    19)"=";:NR = TPAYMENT:DOLL$ = "YES": GOSUB
    9000
1070 VTAB 18: HTAB 2: PRINT "Principal"; TAB(
    19)"=";:NR = LOAN: GOSUB 9000
1080 VTAB 19: HTAB 2: PRINT "Interest"; TAB(
    19)"=";:NR = TPAYMENT - LOAN: GOSUB 90
    00
1090 VTAB 22: HTAB 1: PRINT L$
1100 VTAB 23: HTAB 14: PRINT "Press any ke
    y ";
1110 GET S$
1120 RETURN
1130 REM PAYMENTS PER PERIOD
1140 IF CHOICE = 1 THEN P$ = "YR":T$ = "YE
    ARLY PAYMENT EQUALS"
1150 IF CHOICE = 2 THEN P$ = "YR:QT":T$ =
    "QUARTERLY PAYMENT EQUALS"
1160 IF CHOICE = 3 THEN P$ = "YR:MT":T$ =
    "MONTHLY PAYMENT EQUALS"
1170 YR = 1:QT = 1:MT = 1
1180 FOR I = 1 TO N STEP 12
1190 REM HEADING
1200 GOSUB 1250
1210 REM BODY
1220 GOSUB 1340
1230 NEXT I
1240 RETURN
1250 REM HEADING
1260 HOME
1270 PRINT L$

```

```

1280 PRINT TAB( 20 - LEN (T$) / 2);T$
1290 HT = 23:NR = PPP: GOSUB 9000
1300 PRINT L$
1310 PRINT TAB( 16)"Paid to"; TAB( 33)"Pa
    id to"
1320 PRINT P$; TAB( 15)"Principal"; TAB( 3
    2)"Interest
1330 RETURN
1340 REM BODY
1350 ROW = 8
1360 FOR J = 1 TO I + 11
1370 IF J > N THEN 1490
1380 PTI = R * LOAN
1390 PRN = PPP - PTI
1400 VTAB ROW: HTAB 1: PRINT YR;
1410 IF CHOICE = 2 THEN PRINT ":";QT
1420 IF CHOICE = 3 THEN PRINT ":";MT
1430 VTAB ROW:HT = 23:NR = PRN: GOSUB 9000
1440 VTAB ROW:HT = 39:NR = PTI: GOSUB 9000
1450 REM FREQUENCY
1460 ON CHOICE GOSUB 1540,1570,1610
1470 ROW = ROW + 1
1480 LOAN = LOAN - PRN
1490 NEXT J
1500 VTAB 22: HTAB 1: PRINT L$
1510 VTAB 23: HTAB 14: PRINT "Press any ke
    y ";
1520 GET S$
1530 RETURN
1540 REM YEAR
1550 YR = YR + 1
1560 RETURN
1570 REM QUARTER
1580 QT = QT + 1
1590 IF QT = 5 THEN ROW = ROW + 1:YR = YR +
    1:QT = 1
1600 RETURN
1610 REM MONTH
1620 MT = MT + 1
1630 IF MT = 13 THEN YR = YR + 1:MT = 1
1640 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
    00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"

```

```

9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

College Planner

College costs are spiraling ever upward. Tuition with room and board at many state-supported schools often runs a couple of thousand dollars a year, and double or triple that at private schools. With costs like these, a long-range savings plan is certainly useful. That's where "College Planner" comes in.

If you have children, you're probably assuming they'll go on to some form of higher education. Perhaps they'll take classes at a community college. Maybe at the local technical school. Or go on to a four-year university. All cost money. How are you going to save for those years?

"College Planner" gives you an idea of what it will cost to send a child beyond high school. As with the other financial programs in this chapter, you can run through it as many times as you want, changing the parameters to reflect different situations. This will give you a better idea of what it *might* cost to send your child to college.

The Paper Chase

After typing in and saving College Planner, run it. Unlike the previous programs, you have to select the appropriate menu if you want to change the default parameters (the values set when the program first runs). It's not difficult.

The easiest way to show how College Planner works is to go through an example.

Let's assume your child is now 12 years old. The first display screen asks for the number of years until the paper chase begins. Respond with 6, since $18 - 12 = 6$.

The computer then displays the main menu. Here, you decide if you want to review savings, expenses, or economic assumptions, compute totals, or exit the program. Pressing a single key selects another menu. Figure 1 shows the options available in the main menu.

Review savings. By pressing the 1 key, you'll see a menu on the screen. Notice the annual savings default settings of \$1,000 for the parents and \$500 for the child. Just press the Y key to change either of these. You'll be asked for the item to

change (A or B), and then the new amount. Let's change them to \$500 for the parents' and \$250 for the child's contribution. After you've entered the figures, press the N key and you'll return to the main menu.

Figure 1. Menu for College Planner

Would you like to

- ☐ 1 Review savings
- ☐ 2 Review expenses
- ☐ 3 Review economic assumptions
- ☐ 4 Compute totals
- ☐ 5 Exit

Choice = ?

Review expenses. Now press the 2 key. You should see another menu at this time. Again, to change the amount in each category, press the Y key, select the item, and enter the new figure. For example, let's say that your child is going to a school several states away, with higher travel expenses to and from school. Change that category to \$500. Press the N key to return to the main menu.

Review economic assumptions. Press the 3 key and you'll see the display change. The default values of 10 percent for expected interest rate and 5 percent for expected inflation rate can be changed. The interest rate is what your yearly savings will be compounded by, while the inflation rate signifies how much costs increase each year. College Planner assumes that college costs increase at the same rate as inflation. This might not be true in a single year, but over a longer period of time, it's accurate enough for this forecaster. Let's change the interest rate to 12 percent. (You were able to make better-than-average investments.) Hit the N key to return to the main menu.

Compute totals. Hit the 4 key and the program will compute what it's going to cost you to send your child to college. The first screen displays the cost (based on what you entered in the *Review expenses* menu) of the education when you

child reaches college age. The numbers are adjusted for inflation, and in fact increase each year by 5 percent, the rate of inflation you earlier specified. Note that these figures are in *future dollars*—1991 dollars in the example (remember, you said it was six years until the start of school). The first year will cost \$5,100, the second year \$5,355, and so on. Press any key and the Bottom Line screen displays as in Figure 2.

Figure 2. The Bottom Line

The Bottom Line		
SAVINGS FOR COLLEGE		
Principal	=	\$ 7,500.00
Interest	=	\$ 7,240.94
Total	=	\$14,740.94
COST OF COLLEGE		
In today's dollars	=	\$14,500.00
In inflated dollars	=	\$21,984.00
SAVINGS minus INFLATED COST	=	\$-7,243.89

This shows how much you saved, the interest you received over those six years, the total money available, the cost of college in today's dollars, the total in inflated dollars, and the final balance. In the example, note that you're in the red by over \$7,000. You obviously didn't save enough. Hitting any key will return you to the main menu again.

Now you can run the program again, this time entering a higher amount of savings. By trial and error, you'll find how much you'll have to save over the next six years in order to pay for your child's education. Of course, when you use this program yourself, you'll be entering different parameters to reflect your child's age and your own estimates of what college will cost.

College Planner

```

100 REM COLLEGE PLANNER
110 REM INITIALIZE
120 GOSUB 200
130 REM ENTER VALUES
    
```



```

140 GOSUB 600
150 REM CHOOSE FROM MAIN MENU
160 GOSUB 770
170 ON PICK GOSUB 900,900,900,1290
180 IF PICK < > 5 THEN 160
190 END
200 REM INITIALIZE
210 REM TITLE
220 GOSUB 280
230 REM HEADING
240 GOSUB 360
250 REM KEY VALUES
260 GOSUB 470
270 RETURN
280 REM TITLE
290 PRINT CHR$(21): TEXT : HOME
300 VTAB 13: HTAB 12: PRINT "College Plann
er";
310 FOR PAUSE = 1 TO 1500: NEXT
320 L$ = " ": FOR I = 1 TO 39: L$ = L$ + "=":
NEXT
330 REM DIGITS AFTER DECIMAL
340 DA = 2
350 RETURN
360 REM HEADING
370 HOME
380 PRINT "College Planner is designed to
help you
390 PRINT "develop a plan to pay for your
child's
400 PRINT "education.
410 PRINT
420 PRINT "College Planner assumes that yo
ur
430 PRINT "annual savings will earn intere
st but
440 PRINT "that inflation will increase yo
ur
450 PRINT "expenses.
460 RETURN
470 REM KEY VALUES
480 REM NUMBER OF CATEGORIES
490 K = 3
500 REM MAX NUMBER OF ITEMS PER CATEGORY
510 DATA 10
520 READ NX
530 DIM C$(3),EF(NX + 1,4),X$(NX,3),X(NX,3
)
540 REM ACTUAL NUMBER OF ITEMS
550 DATA 2,6,2

```

```

560  FOR I = 1 TO K
570  READ N(I)
580  NEXT
590  RETURN
600  REM  ENTER VALUES
610  REM  YEARS TO COLLEGE
620  VTAB 10: HTAB 1: PRINT "How many years
    are there until college
630  VTAB 11: HTAB 10: PRINT  SPC( 15); CHR$
    (7)
640  VTAB 11: HTAB 1: INPUT "begins ? ";S$
650  N = VAL (S$)
660  IF N < 1 THEN  VTAB 23: HTAB 3: PRINT
    "It's a bit too late for planning !": GOTO
    630
670  REM  CATEGORIES
680  FOR I = 1 TO K
690  READ C$(I)
700  NEXT
710  REM  ITEMS
720  FOR I = 1 TO K
730  FOR J = 1 TO N(I)
740  READ X$(J,I),X(J,I)
750  NEXT J,I
760  RETURN
770  REM  MAIN MENU
780  HOME
790  VTAB 6: HTAB 10: PRINT "Would you like
    to
800  VTAB 8: HTAB 11: INVERSE : PRINT "1";:
    NORMAL : PRINT " Review savings
810  VTAB 10: HTAB 11: INVERSE : PRINT "2";
    : NORMAL : PRINT " Review expenses
820  VTAB 12: HTAB 11: INVERSE : PRINT "3";
    : NORMAL : PRINT " Review economic ass
    umptions
830  VTAB 14: HTAB 11: INVERSE : PRINT "4";
    : NORMAL : PRINT " Compute totals
840  VTAB 16: HTAB 11: INVERSE : PRINT "5";
    : NORMAL : PRINT " Exit
850  VTAB 18: HTAB 10: PRINT "Choice = ? ";
    CHR$ (7);
860  GET S$
870  PICK = VAL (S$)
880  IF PICK < 1 OR PICK > 5 THEN 850
890  RETURN
900  REM  UPDATE
910  P = PICK
920  Z = N(P)
930  REM  DISPLAY

```

```

940 GOSUB 980
950 REM SELECT
960 GOSUB 1110
970 RETURN
980 REM DISPLAY
990 HOME
1000 PRINT L$
1010 PRINT TAB( 20 - LEN (C$(P)) / 2)C$(
P)
1020 PRINT L$
1030 IF PICK < > 3 THEN DOLL$ = "YES"
1040 FOR J = 1 TO Z
1050 VTAB 4 + J: HTAB 1: INVERSE : PRINT CHR$
(J + 64);: NORMAL
1060 PRINT TAB( 3)X$(J,P); TAB( 22)"=";:H
T = 37:NR = X(J,P): GOSUB 9000
1070 NEXT J
1080 DOLL$ = ""
1090 VTAB 17: HTAB 1: PRINT L$
1100 RETURN
1110 REM SELECT
1120 VTAB 18: HTAB 1: PRINT "Change value
(Y/N) ? "; CHR$ (7);
1130 GET S$
1140 IF S$ = "N" OR S$ = "n" THEN 1280
1150 IF S$ < > "Y" AND S$ < > "y" THEN 1
120
1160 VTAB 20: HTAB 1: PRINT "Item = ? "; CHR$
(7);
1170 GET S$
1180 A = ASC (S$)
1190 IF A > 90 THEN A = A - 32
1200 REM ELEMENT IN VECTOR
1210 Q = A - 64
1220 IF Q < 1 OR Q > Z THEN 1160
1230 VTAB 20: HTAB 10: PRINT CHR$ (A)
1240 REM NEW VALUE
1250 VTAB 22: HTAB 1: PRINT CHR$ (7);: INPUT
"New value = ? ";S$
1260 X(Q,P) = VAL (S$)
1270 GOSUB 980: GOTO 1120
1280 RETURN
1290 REM COMPUTE TOTALS
1300 HOME
1310 VTAB 12: HTAB 15: FLASH : PRINT "COMP
UTING": NORMAL
1320 REM SAVINGS
1330 GOSUB 1430
1340 REM EXPENSES
1350 GOSUB 1620

```

```

1360 REM TOTAL EXPENSES
1370 GOSUB 1710
1380 REM DISPLAY EXPENSES
1390 GOSUB 1860
1400 REM DISPLAY TOTALS
1410 GOSUB 2110
1420 RETURN
1430 REM SAVINGS
1440 T = N + 4
1450 REM INTEREST RATE IN INDEX FORM
1460 RT = 1 + X(1,3) / 100
1470 REM SAVINGS PER YEAR
1480 SPY = 0
1490 FOR J = 1 TO N(1)
1500 SPY = SPY + X(J,1)
1510 NEXT
1520 REM PRINCIPAL
1530 SAVPRN = SPY * T
1540 REM TOTAL (INCLUDES INTEREST)
1550 ST = 0
1560 FOR I = 1 TO T
1570 ST = ST + SPY * RT ^ (T - I + 1)
1580 NEXT
1590 REM INTEREST
1600 SI = ST - SAVPRN
1610 RETURN
1620 REM EXPENSES
1630 REM INFLATION RATE IN INDEX FORM
1640 RT = 1 + X(2,3) / 100
1650 Q = N(2)
1660 FOR I = 1 TO Q
1670 FOR J = 1 TO 4
1680 EF(I,J) = X(I,2) * RT ^ (N + J)
1690 NEXT J,I
1700 RETURN
1710 REM TOTALS
1720 FOR I = 1 TO 4
1730 EF(Q + 1,I) = 0
1740 FOR J = 1 TO Q
1750 EF(Q + 1,I) = EF(Q + 1,I) + EF(J,I)
1760 NEXT J,I
1770 REM GRAND TOTALS
1780 CK = 0:CF = 0
1790 FOR I = 1 TO 4
1800 CF = CF + EF(Q + 1,I)
1810 NEXT
1820 FOR I = 1 TO Q
1830 CK = CK + X(I,2) * 4
1840 NEXT
1850 RETURN

```

```

1860 REM DISPLAY
1870 X$(Q + 1,2) = "Total
1880 YR$(1) = "1ST":YR$(2) = "2ND":YR$(3) =
      "3RD":YR$(4) = "4TH"
1890 DOLL$ = "YES"
1900 FOR I = 1 TO 4
1910 GOSUB 1940
1920 NEXT I
1930 RETURN
1940 REM HEADING
1950 HOME
1960 PRINT L$
1970 PRINT TAB( 3)"COLLEGE EXPENSES: INFL
      ATED DOLLARS
1980 PRINT L$
1990 VTAB 5: HTAB 1: INVERSE : PRINT "CATE
      GORY"; HTAB 28: PRINT YR$(I);" YEAR":
      NORMAL
2000 PRINT
2010 REM BODY
2020 FOR J = 1 TO Q + 1
2030 PRINT X$(J,2); TAB( 20)"=";
2040 HT = 35:NR = EF(J,I): GOSUB 9000
2050 IF J = Q THEN PRINT
2060 NEXT J
2070 VTAB 22: HTAB 1: PRINT L$
2080 VTAB 23: HTAB 14: PRINT "Press any ke
      y ";
2090 GET S$
2100 RETURN
2110 REM TOTALS
2120 HOME
2130 PRINT L$
2140 PRINT TAB( 13)"THE BOTTOM LINE
2150 PRINT L$
2160 INVERSE : PRINT "SAVINGS FOR COLLEGE"
      : NORMAL
2170 PRINT
2180 PRINT "Principal"; TAB( 21)"=";:HT =
      36:NR = SAVPRN: GOSUB 9000
2190 PRINT "Interest"; TAB( 21)"=";:NR = S
      I: GOSUB 9000
2200 PRINT "Total"; TAB( 21)"=";:NR = ST: GOSUB
      9000
2210 VTAB 12: HTAB 1: INVERSE : PRINT "COS
      T OF COLLEGE": NORMAL
2220 PRINT
2230 PRINT "In today's dollars"; TAB( 21)"
      =" ;:NR = CK: GOSUB 9000

```

```

2240 PRINT "In inflated dollars"; TAB( 21)
    "=";:NR = CF: GOSUB 9000
2250 VTAB 18: HTAB 1: INVERSE : PRINT "SAV
INGS MINUS"
2260 PRINT "INFLATED COST";: NORMAL : PRINT
    TAB( 21)"=";:NR = ST - CF: GOSUB 9000
2270 VTAB 22: HTAB 1: PRINT L$
2280 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
2290 GET S$
2300 RETURN
2310 REM DATA
2320 REM CATEGORIES
2330 DATA Annual Savings, First-Year Expe
nses (Today's prices), Economic Assump
tions
2340 REM SAVINGS
2350 DATA Parents, 1000
2360 DATA Kids - summer jobs, 500
2370 REM EXPENSES
2380 DATA Tuition, 1500
2390 DATA Room & Board, 1000
2400 DATA Books, 300
2410 DATA Travel To School, 50
2420 DATA Laundry, 75
2430 DATA Entertainment, 250
2440 REM ECONOMIC ASSUMPTIONS
2450 DATA Interest rate (%),10
2460 DATA Inflation rate (%),5
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Effective Yield on Investments

"Money has value over time." Interest on your investment can be compounded with almost any frequency. This simple program shows you which frequency gives you the best return.

Whenever you make a deposit in a savings account, you're investing. You deposit \$500, for example, only on condition that you get back more than that in the future. It doesn't make sense to merely accept what you put in, because inflation, no matter how low, is constantly eroding the purchasing power of your money.

So you deposit \$500 today in return for \$500 plus 10 percent interest a year from now. This interest rate, or the price of money, links dollars today with those of tomorrow. Your \$500, then, is worth \$550 in one year— $\$500 * (1 + 0.10)$ —if interest is compounded annually.

Interest, however, may be compounded with almost any frequency—semiannually, quarterly, monthly, daily, and even continuously are all typical examples from today's complicated business world.

When interest is compounded more than once a year, *nominal* and *effective* interest rates differ. For example, a \$500 investment, with interest compounded twice yearly at a 10 percent *nominal* rate, yields $\$500 * (1 + 0.10/2)^2$ in a year, or \$551.25. The *effective* interest rate, or the actual percentage return on your money, is thus 10.25 percent per annum.

With this short program, "Effective Yield on Investments," you can discover how different interest compounding can affect your money. How can you best decide where to put your money unless you know what the result will be? This program helps you find out.

Compounding Frequency

It won't take long to type in and save Effective Yield on Investments. Once you've entered it, type RUN. There's only one piece of information you have to provide—the nominal interest rate. If that's 12 percent, type in 12 and hit the RETURN key. The program does the rest.

Your computer calculates a host of effective interest rates for any nominal rate you've entered, as the table shows. Using the nominal rate of 12 percent, the figure shows what you'd see on your screen.

Effective Yields (Nominal Rate = 12%)

Frequency of Compounding	Effective Yield, %
Annual	12.000
Twice yearly	12.360
Quarterly	12.551
Monthly	12.683
Daily	12.747
Continuously	12.750

This will help you determine the exact return on a proposed investment. If, for example, \$10,000 is deposited at a 12 percent nominal rate, compounded annually, you'll have \$11,200 at the end of one year. That's because $\$10,000 * (1 + 0.12) = \$11,200$. Continuous compounding, on the other hand, would yield \$11,275 ($\$10,000 * (1 + 0.1275)$), or an additional \$75.

Effective Yield on Investments

```

100 REM EFFECTIVE YIELD ON INVESTMENTS
110 REM INITIALIZE
120 GOSUB 200
130 REM ENTER NOMINAL YIELD
140 GOSUB 330
150 REM COMPUTE EFFECTIVE YIELD
160 GOSUB 580
170 REM DISPLAY RESULTS
180 GOSUB 680
190 END
200 REM INITIALIZE
210 PRINT CHR$(21): TEXT : HOME
220 PRINT "This program computes effective
    yields
230 PRINT "on an investment.
240 PRINT
250 PRINT "It does this for various frequencies of
    interest compounding.
260 PRINT
270 L$ = "": FOR I = 1 TO 39:L$ = L$ + " ":
    NEXT

```



```

280 REM DIGITS AFTER DECIMAL
290 DA = 3
300 VTAB 23: HTAB 14: PRINT "Press any key .
    ";
310 GET S$
320 RETURN
330 REM ENTER NOMINAL YIELD
340 REM FREQUENCIES
350 GOSUB 390
360 REM ENTRY
370 GOSUB 450
380 RETURN
390 REM FREQUENCIES
400 DATA Annual, Twice yearly, Quarterly,
    Monthly, Daily, Continuously
410 FOR I = 1 TO 6
420 READ FQ$(I)
430 NEXT
440 RETURN
450 REM ENTRY
460 HOME
470 PRINT "Please enter the nominal yield
    of your
480 PRINT "investment.
490 PRINT
500 PRINT "For example, if the yield is 7%
    a year,
510 PRINT "enter 7. If it's 12% a year, e
    nter 12,
520 PRINT "and so on.
530 VTAB 8: HTAB 11: PRINT SPC( 15); CHR$
    (7)
540 VTAB 8: HTAB 1: INPUT "Yield = ? ";Y$
550 NOMYD = VAL (Y$)
560 IF NOMYD < = 0 THEN VTAB 23: HTAB 9:
    PRINT "There's no free lunch !"; GOTO
    530
570 RETURN
580 REM COMPUTE
590 NOMYD = NOMYD / 100
600 DEF FN YD(F) = (((1 + NOMYD / F) ^ F)
    - 1) * 100
610 YIELD(1) = FN YD(1)
620 YIELD(2) = FN YD(2)
630 YIELD(3) = FN YD(4)
640 YIELD(4) = FN YD(12)
650 YIELD(5) = FN YD(365)
660 YIELD(6) = ( EXP (NOMYD) - 1) * 100
670 RETURN
680 REM DISPLAY

```

```

690 HOME
700 PRINT L$
710 PRINT TAB( 10)"NOMINAL YIELD = ";NOMY
    D * 100; "%"
720 PRINT L$
730 VTAB 5: HTAB 6: INVERSE : PRINT "FREQU
    ENCY OF";: HTAB 28: PRINT "EFFECTIVE
740 VTAB 6: HTAB 6: PRINT "COMPOUNDING ";:
    HTAB 30: PRINT "YIELD": NORMAL
750 ROW = 9:HT = 34
760 FOR I = 1 TO 6
770 VTAB ROW: HTAB 6: PRINT FQ$(I);:NR = Y
    IELD(I): GOSUB 9000
780 VTAB ROW: HTAB 36: PRINT "%"
790 ROW = ROW + 2
800 NEXT
810 VTAB 22: HTAB 1: PRINT L$
820 VTAB 23: HTAB 14: PRINT "Press any key
    ";
830 GET S$
840 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
    00000"
9020 N$ = STR$(NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
    1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
    L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$(Z
    Z$ + STR$(ZR),DA)
9080 ZT$ = " " + STR$(ZL)
9090 IF MID$(ZT$, LEN (ZT$) - 3,1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$(ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Municipal Bond Buyer

Municipal bonds are becoming more and more attractive as investments, for they not only yield returns competitive with money market certificates, but are also exempt from federal taxes. No wonder so many people have turned to them. Using this program, you can chart the return on any municipal bond, a great aid when you're trying to decide if you want to buy.

If you're like most people, you cringe every April 15, especially if you have a lot of interest income. You've tried to save, but so much of the interest you've earned seems to end up with the Internal Revenue Service. If you're in a 30 percent tax bracket, for example, the IRS takes \$300 in taxes out of every \$1,000 of interest you've earned. It hurts. There has to be a more effective way of maintaining your investments.

Fortunately, there is. Municipal bonds offer rates of return highly competitive with money market certificates; they're also free from federal income taxes and even some state taxes. Therefore, a 9 percent return on a municipal bond is equivalent to a 13.5 percent return on a taxable investment, if you're in the 25 percent tax bracket. No wonder sales of municipal bonds have been increasing over the past few years.

A municipal bond is a "promise to pay" issued by a state, city, county, town, housing authority, university, or other administrative body. The organization sells a bond and promises to pay the principal back at a later date, say, in 20 or 30 years, with a fixed amount of interest usually paid every 6 months.

Municipal bonds finance the construction of public works projects like schools, roads, sewers, nursing homes, hospitals, and low-income housing. The bond issuers do not use current tax revenues to pay for these projects simply because many of the projects will last a lifetime, and it's only right that future generations pay their fair share of the cost. Also, it is generally easier politically to issue bonds than to raise taxes.

You can calculate key municipal bond values by using "Municipal Bond Buyer." All you have to do is type in the program, save it, and run it.

Falls River

To illustrate Municipal Bond Buyer, let's go through an example. The city of Falls River decides to build a sewer. Construction is financed through the sale of Falls River bonds, with each bond carrying a coupon rate of 10 percent and a face value of \$1,000. To simulate this, at the first screen display, enter 10 for the coupon rate.

The bonds were issued in January of 1983 and mature in November of 2004. Interest was paid in June 1983, and every six months thereafter. You bought ten \$1,000 slices of the Falls River issue in March 1983, at a premium price of \$103. This price, notice, is in relation to a *base price*, or *par value*, of \$100. In other words, you're paying a premium of \$3.

To reflect this, you need to enter several things as the program prompts you. The second screen display asks for the price of the bond. When you buy municipal bonds, you usually buy them at a discount, especially if they've just been issued. At other times, however, you'll buy them at a premium, simply because of the attractive coupon or interest rate. Enter 103 at the second screen display; this indicates that you've paid \$103 for a par value of \$100.

The next screen asks for the number of \$1,000 slices you want to buy. Type in 10 and press RETURN. Another prompt displays and you're asked for the last time interest was paid on the bond. (If you bought a newly issued bond, enter the date of purchase.) Interest hasn't yet been paid on these bonds, so type in 1983 and 1. Now enter the year and month you bought the bond; that would be 1983 and 3, for March of that year. The bonds mature in November of the year 2004; type in 2004 and 11 at the next prompt.

The program also asks for your tax bracket. That's the percentage of your income that would go for taxes if you had no deductions. If you file a joint return and your gross income is between \$29,000 and \$35,200, for example, you're roughly in the 30 percent tax range. Enter 30 for this example. Remember, all of this is hypothetical. No one knows what changes will be made in the tax laws by the time you read this. You can find out what bracket you're in by calling the Internal Revenue Service.

Now, the bond summary appears on the screen. This is the heart of the program. The total cost of your purchase was \$10,466.67. Of this amount, \$10,300.00 is paid to principal.

That is, 10 bonds * \$1,000 per bond * 1.03 (price index) = \$10,300.00. An additional \$166.67 is paid to interest.

The interest payment on a bond is *always* fixed. In the Falls River case, you received \$500.00 in June 1983, \$500.00 in December 1983, \$500.00 in June 1984, and so on. But you purchased the bonds in March 1983. Hence, you owed the seller two months worth of interest, for January and February, or 2 * \$500/6 (the monthly interest payment). That's \$166.67.

At the bottom of the display, you'll see that you'll earn \$21,833.33 in tax-free interest over 21 years and 8 months, the life of the bond. Added to the principal of \$10,000.00, your total income will be \$31,833.33. Press any key to continue.

The annual current yield on your investment is 9.71 percent, or the coupon rate divided by the price ($10/103 = 0.0971$). Assuming a 30 percent tax bracket, the equivalent yield on a *taxable* security is an impressive 13.87 percent per annum. That's the interest you'd have to be getting from a taxable investment in order to equal your return on these mythical Falls River bonds. In light of this, it's not surprising that municipal bonds are becoming so popular.

Check It Out

Municipal Bond Buyer is only a guide to help you in your decision making. By using this program, you can compare different bonds, their coupon rates, discount or premium prices, and their times of maturity. But *you* have to make the final choice of which bond to buy, or whether to buy at all. You'll probably want to consult an accountant if things get involved. But at least you'll have an idea of what a bond can do for your investment income, and whether it's something worthwhile to try.

Municipal Bond Buyer

```

100 REM MUNICIPAL BOND BUYER
110 REM HEADING
120 GOSUB 200
130 REM ENTER DATA
140 GOSUB 280
150 REM COMPUTE
160 GOSUB 1430
170 REM DISPLAY RESULTS
180 GOSUB 1790
190 END
200 REM HEADING
    
```

```

210 PRINT CHR$(21): TEXT : HOME
220 VTAB 12: HTAB 15: PRINT "Municipal
230 PRINT TAB(16)"Bond
240 PRINT TAB(17)"Buyer
250 FOR PAUSE = 1 TO 1500: NEXT
260 L$ = "": FOR I = 1 TO 39: L$ = L$ + "=":
    NEXT
270 RETURN
280 REM ENTER DATA
290 REM COUPON RATE
300 GOSUB 400
310 REM PURCHASE PRICE
320 GOSUB 520
330 REM NUMBER OF UNITS PURCHASED
340 GOSUB 680
350 REM DATES
360 GOSUB 760
370 REM TAX BRACKET
380 GOSUB 1290
390 RETURN
400 REM COUPON RATE
410 HOME
420 PRINT "Please enter the coupon rate on
    your
430 PRINT "bond.
440 PRINT
450 PRINT "For example, enter 7 for 7%, 10
    for
460 PRINT "10%, and so on.
470 VTAB 7: HTAB 10: PRINT SPC(15); CHR$(
    7)
480 VTAB 7: HTAB 1: INPUT "Rate = ? ";R$
490 RT = VAL(R$)
500 IF RT < = 0 THEN VTAB 23: HTAB 9: PRINT
    "There's no free lunch !": GOTO 470
510 RETURN
520 REM PURCHASE PRICE
530 HOME
540 PRINT "Please enter the price of your
    bond.
550 PRINT "Use $100 as a base figure, or '
    par'
560 PRINT "value.
570 PRINT
580 PRINT "For example, enter a number lik
    e $95
590 PRINT "($5.00 discount).
600 PRINT
610 PRINT "Or enter a number like $103.75,
620 PRINT "($3.75 premium), and so on.

```

```

630 VTAB 11: HTAB 11: PRINT SPC( 15); CHR$
    (7)
640 VTAB 11: HTAB 1: INPUT "Price = ? ";P$
650 PRICE = VAL (P$)
660 IF PRICE < = 0 OR PRICE > 200 THEN VTAB
    23: HTAB 9: PRINT "Par Value is $100.0
    0 !"; GOTO 630
670 RETURN
680 REM PURCHASE
690 HOME
700 PRINT "How many $1,000 slices of the b
    ond do
710 VTAB 2: HTAB 19: PRINT SPC( 15); CHR$
    (7)
720 VTAB 2: HTAB 1: INPUT "you want to buy
    ? ";Q$
730 QTY = INT ( VAL (Q$))
740 IF QTY < = 0 THEN VTAB 23: HTAB 9: PRINT
    "Please buy something !": GOTO 710
750 RETURN
760 REM DATES
770 REM LAST INTEREST PAYMENT
780 GOSUB 840
790 REM PURCHASE
800 GOSUB 1090
810 REM MATURITY
820 GOSUB 1190
830 RETURN
840 REM LAST INTEREST PAYMENT
850 HOME
860 PRINT "Please enter the date that inte
    rest was
870 PRINT "last paid on your bond.
880 PRINT
890 PRINT "If interest has not yet been pa
    id,
900 PRINT "don't worry. Enter the date th
    at the
910 PRINT "bond was issued instead.
920 GOSUB 950
930 MT(1) = MT:YR(1) = YR
940 RETURN
950 REM DATES
960 REM YEAR
970 VTAB 10: HTAB 1: PRINT SPC( 40)
980 VTAB 8: HTAB 11: PRINT SPC( 15); CHR$
    (7)
990 VTAB 8: HTAB 2: INPUT "Year = ? ";Y$
1000 YR = INT ( VAL (Y$))

```

```

1010 IF LEN ( STR$ (YR)) < > 4 THEN VTAB
      23: HTAB 5: PRINT "Please enter a 4-di
      git year !": GOTO 980
1020 VTAB 23: HTAB 5: PRINT SPC( 35)
1030 REM MONTH
1040 VTAB 10: HTAB 11: PRINT SPC( 15); CHR$
      (7)
1050 VTAB 10: HTAB 1: INPUT "Month = ? ";M
      $
1060 MT = INT ( VAL (M$))
1070 IF MT < 1 OR MT > 12 THEN 1040
1080 RETURN
1090 REM PURCHASE DATE
1100 HOME
1110 PRINT "Please enter the date that you
1120 PRINT "purchased the bond.
1130 GOSUB 950
1140 MT(2) = MT:YR(2) = YR
1150 REM MONTHS BETWEEN DATES
1160 N1 = (YR(2) - YR(1)) * 12 + MT(2) - MT
      (1)
1170 IF N1 < 0 THEN VTAB 23: HTAB 6: PRINT
      "Please enter a later date !": GOTO 11
      30
1180 RETURN
1190 REM MATURITY
1200 HOME
1210 PRINT "Please enter the date that you
      r bond
1220 PRINT "matures.
1230 GOSUB 950
1240 MT(3) = MT:YR(3) = YR
1250 REM MONTHS BETWEEN DATES
1260 N2 = (YR(3) - YR(2)) * 12 + MT(3) - MT
      (2)
1270 IF N2 < 0 THEN VTAB 23: HTAB 6: PRINT
      "Please enter a later date !": GOTO 12
      30
1280 RETURN
1290 REM TAX BRACKET
1300 HOME
1310 PRINT "Please enter your marginal tax
      bracket."
1320 PRINT
1330 PRINT "For example, enter 33 if you'r
      e in the
1340 PRINT "33% bracket.
1350 PRINT
1360 PRINT "And enter 50 if you're in the
      50%

```



```

1370 PRINT "bracket, and so on.
1380 VTAB 9: HTAB 13: PRINT SPC( 15); CHR#
(7)
1390 VTAB 9: HTAB 1: INPUT "Bracket = ? ";
B$
1400 BK = VAL (B$)
1410 IF BK < = 0 OR BK > 50 THEN 1380
1420 RETURN
1430 REM COMPUTE
1440 REM TOTAL COST
1450 GOSUB 1530
1460 REM TIME TO MATURITY
1470 GOSUB 1640
1480 REM TOTAL PAYMENT
1490 GOSUB 1680
1500 REM YIELDS
1510 GOSUB 1730
1520 RETURN
1530 REM COST
1540 RT = RT / 100
1550 REM MONTHLY INTEREST PAYMENT
1560 PM = QTY * 1000 * RT / 12
1570 REM INTEREST COST
1580 CI = PM * N1
1590 REM PRINCIPAL COST
1600 CP = QTY * PRICE * 10
1610 REM NET
1620 CN = CI + CP
1630 RETURN
1640 REM TIME TO MATURITY
1650 YEARS = INT (N2 / 12)
1660 MTHS = N2 - YEARS * 12
1670 RETURN
1680 REM TOTAL PAYMENT
1690 PP = QTY * 1000
1700 PI = PP * RT * (N1 + N2) / 12
1710 PN = PP + PI
1720 RETURN
1730 REM YIELDS
1740 REM TAX-FREE
1750 CY = RT * 100 / PRICE
1760 REM TAXABLE
1770 TY = CY / (1 - BK / 100)
1780 RETURN
1790 REM DISPLAY RESULTS
1800 REM HEADING
1810 GOSUB 1900
1820 REM BODY
1830 GOSUB 1960
1840 VTAB 22: HTAB 1: PRINT L$

```

```

1850 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
1860 GET S$
1870 REM YIELDS
1880 GOSUB 2280
1890 RETURN
1900 REM HEADING
1910 HOME
1920 PRINT L$
1930 PRINT TAB( 9)"MUNICIPAL BOND SUMMARY
1940 PRINT L$
1950 RETURN
1960 REM BODY
1970 REM BOND DESCRIPTION
1980 GOSUB 2040
1990 REM COST
2000 GOSUB 2160
2010 REM INCOME
2020 GOSUB 2220
2030 RETURN
2040 REM DESCRIPTION
2050 VTAB 5: HTAB 1: PRINT "Coupon Rate"; TAB(
15)"=";;DA = 2:HT = 30:NR = RT * 100: GOSUB
9000
2060 VTAB 5: HTAB 32: PRINT "%"
2070 VTAB 6: HTAB 1: PRINT "Bond Price"; TAB(
15)"=";;DOLL$ = "YES":NR = PRICE: GOSUB
9000
2080 DT$ = STR$ (MT(1)) + "/" + STR$ (YR(
1))
2090 VTAB 8: HTAB 1: PRINT "Interest Date"
; TAB( 15)"="; TAB( 31 - LEN (DT$));D
T$
2100 DT$ = STR$ (MT(2)) + "/" + STR$ (YR(
2))
2110 VTAB 9: HTAB 1: PRINT "Purchase Date"
; TAB( 15)"="; TAB( 31 - LEN (DT$));D
T$
2120 DT$ = STR$ (MT(3)) + "/" + STR$ (YR(
3))
2130 VTAB 10: HTAB 1: PRINT "Maturity Date
"; TAB( 15)"="; TAB( 31 - LEN (DT$));
DT$
2140 VTAB 11: HTAB 1: PRINT "Length"; TAB(
15)"= ";YEARS;" Years and ";MTHS;" Mon
ths
2150 RETURN
2160 REM COST
2170 VTAB 13: HTAB 1: INVERSE : PRINT "TOT
AL COST": NORMAL
2180 VTAB 14: HTAB 1: PRINT "Principal"; TAB(
15)"=";;NR = CP: GOSUB 9000

```

```

2190 VTAB 15: HTAB 1: PRINT "Interest"; TAB(
15)"=";:NR = CI: GOSUB 9000
2200 VTAB 16: HTAB 1: PRINT "Net"; TAB( 15
)"=";:NR = CN: GOSUB 9000
2210 RETURN
2220 REM INCOME
2230 VTAB 18: HTAB 1: INVERSE : PRINT "TOT
AL INCOME": NORMAL
2240 VTAB 19: HTAB 1: PRINT "Principal"; TAB(
15)"=";:NR = PP: GOSUB 9000
2250 VTAB 20: HTAB 1: PRINT "Interest"; TAB(
15)"=";:NR = PI: GOSUB 9000
2260 VTAB 21: HTAB 1: PRINT "Net"; TAB( 15
)"=";:NR = PN: GOSUB 9000
2270 RETURN
2280 REM YIELDS
2290 HOME
2300 PRINT L$
2310 PRINT TAB( 10)"MUNICIPAL BOND YIELDS
2320 PRINT L$
2330 VTAB 5: HTAB 1: PRINT "Tax-Free"; TAB(
16)"=";:NR = CY * 100:HT = 31: GOSUB 9
000
2340 VTAB 5: HTAB 33: PRINT "%"
2350 VTAB 7: HTAB 1: PRINT "Tax-Equivalent
=";:NR = TY * 100: GOSUB 9000
2360 VTAB 7: HTAB 33: PRINT "%"
2370 VTAB 22: HTAB 1: PRINT L$
2380 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
2390 GET S$
2400 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "- "
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Household Budget

Sometimes it's tough to make ends meet. Maybe you've found yourself in a financial half nelson while waiting for the next paycheck. "Household Budget" can help you organize.

"Household Budget" is designed to help you manage your home finances. By reviewing what you spend your money on, perhaps you can develop a plan for erasing the red ink or for starting a nest egg.

Meet the Goodbuys

Type in and save the program, then enter RUN. You're ready to meet the Goodbuys, a typical American couple who've decided to tighten their belts. It won't be easy. Like many of us, they tend to spend more than they would like.

After running Household Budget, the Goodbuys are asked to input their monthly income. After adding up paychecks, stock dividends, interest payments, and small gifts from parents, they get a total of \$1,500. That's net income, or take-home pay. Help them out in their attempt to balance their budget. Enter 1500 at the first prompt. (The program automatically supplies dollar signs and decimal points.)

The first menu you'll see asks for a choice from three categories. They are:

Review spending. By pressing the 1 key at the main menu, another display appears. Eight expenditures categories are listed, from HOME OR APARTMENT to MISCELLANEOUS. Any of these categories can be called to the screen. Press 1 to see the first category.

A number of items now show on the screen. Mortgage payments, taxes, insurance, repairs, and rent are listed. Notice the default values for each; this is what the Goodbuys's expenses are. To change the values to reflect your own expenses, simply press the Y key, then the item letter. Enter the new value. To return to the expenditures menu, press the N key.

Any or all of the categories can be changed in this same way. Building your own basic budget takes only a short time. Even if you don't change anything at this point, look at each category for a moment just to see what each includes. When

you're through, press the 9 key, which will return you to the main menu.

Display totals. Accessed by pressing the 2 key, this feature calculates the spending totals, as well as shows each category's percentage of the total expenditures. If you didn't change any of the initial values, you can see that the Goodbuys are spending a total of \$2,265.28 each month. Press any key to see the last display. This compares income with expenditures. Notice that the Goodbuys are spending \$765.00 more than they're taking in. Sharp budget cutting is required! Hit any key and you'll see the main menu again.

The Goodbuys—or you—can change any line in the budget without losing other entered data. You can experiment with paring down various items by using this feature. Cut a bit here and a bit there, refigure the bottom line, and continue until you're satisfied.

Exit. Press the 3 key when you see the main menu on the screen and you will exit the program. Of course, you can run it again if you want to change parameters. However, the default values reappear, and you'll need to enter your own income and expenses again.

Customizing

If you would like to change the data in the program itself, you can save time and effort in the future. After modifying the default values, you have fewer changes to make each time you use the program. The DATA statements in lines 2270–2790 contain the default values, and are easy to alter by simply retyping the line you wish to change.

Household Budget

```

100 REM HOUSEHOLD BUDGET
110 REM INITIALIZE
120 GOSUB 200
130 REM ENTER DATA
140 GOSUB 610
150 REM MAIN MENU
160 GOSUB 890
170 ON PICK GOSUB 1000,1650
180 IF PICK < > 3 THEN 160
190 END
200 REM INITIALIZE
210 REM TITLE
220 GOSUB 280
    
```

```

230 REM HEADING
240 GOSUB 330
250 REM KEY VALUES
260 GOSUB 450
270 RETURN
280 REM TITLE
290 PRINT CHR$(21): TEXT : HOME
300 VTAB 12: HTAB 12: PRINT "Household Bud
get
310 FOR PAUSE = 1 TO 2500: NEXT
320 RETURN
330 REM HEADING
340 HOME
350 PRINT "This program is designed to hel
p you
360 PRINT "better manage your home finance
s.
370 PRINT
380 PRINT "By reviewing what you spend you
r money
390 PRINT "on, you can develop a plan for
400 PRINT "balancing your budget or for re
aching
410 PRINT "a savings goal.
420 VTAB 23: HTAB 14: PRINT "Press any key
";
430 GET S$
440 RETURN
450 REM KEY VALUES
460 REM MAX NUMBER OF CATEGORIES & ITEMS
THEREIN
470 DATA 10,25
480 READ KX,NX
490 DIM C$(KX),SPCT(KX),N(KX),X$(NX,KX),X(
NX,KXX),T(KX)
500 REM ACTUAL NUMBERS
510 DATA 8
520 DATA 5,4,7,4,5,6,6,9
530 READ K
540 FOR I = 1 TO K
550 READ N(I)
560 NEXT
570 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
NEXT
580 REM DIGITS AFTER DECIMAL
590 DA = 2
600 RETURN
610 REM ENTER DATA
620 REM INCOME
630 GOSUB 670

```

```

640 REM SPENDING
650 GOSUB 750
660 RETURN
670 REM SPENDING
680 HOME
690 PRINT "Pardon my asking, but what is y
our
700 VTAB 2: HTAB 18: PRINT SPC( 15); CHR$
(7)
710 VTAB 2: HTAB 1: INPUT "monthly income
? ";S$
720 Y = VAL (S$)
730 IF Y < = 0 THEN VTAB 23: HTAB 10: PRINT
"Better sell your Apple !": GOTO 700
740 RETURN
750 REM SPENDING
760 HOME
770 VTAB 12: HTAB 15: FLASH : PRINT "READI
NG": NORMAL
780 FOR PAUSE = 1 TO 500: NEXT
790 REM CATEGORIES
800 FOR I = 1 TO K
810 READ C$(I)
820 NEXT
830 REM ITEMS
840 FOR I = 1 TO K
850 FOR J = 1 TO N(I)
860 READ X$(J,I),X(J,I)
870 NEXT J,I
880 RETURN
890 REM MAIN MENU
900 HOME
910 VTAB 6: HTAB 11: PRINT "Would you like
to
920 VTAB 8: HTAB 12: INVERSE : PRINT "1";:
NORMAL : PRINT " Review spending
930 VTAB 10: HTAB 12: INVERSE : PRINT "2";
: NORMAL : PRINT " Display totals
940 VTAB 12: HTAB 12: INVERSE : PRINT "3";
: NORMAL : PRINT " Exit
950 VTAB 14: HTAB 11: PRINT "Choice = ? ";
CHR$ (7);
960 GET S$
970 PICK = VAL (S$)
980 IF PICK < 1 OR PICK > 3 THEN 950
990 RETURN
1000 REM SPENDING CATEGORY
1010 REM CHOOSE FROM MENU
1020 GOSUB 1070
1030 REM UPDATE

```

```

1040 IF CHOICE < > K + 1 THEN GOSUB 1220
1050 IF CHOICE < > K + 1 THEN 1020
1060 RETURN
1070 REM MENU
1080 HOME
1090 PRINT "Please enter the number of the
      spending
1100 PRINT "category that you would like t
      o review.
1110 ROW = 4
1120 FOR I = 1 TO K
1130 VTAB ROW: HTAB 10: INVERSE : PRINT I;
      : NORMAL : PRINT " ";C$(I)
1140 ROW = ROW + 2
1150 NEXT
1160 VTAB ROW: HTAB 10: INVERSE : PRINT K +
      1;: NORMAL : PRINT " None";ROW = ROW +
      2
1170 VTAB ROW: HTAB 10: PRINT "Choice = ?
      "; CHR$(7);
1180 GET S$
1190 CHOICE = VAL (S$)
1200 IF CHOICE < 1 OR CHOICE > K + 1 THEN
      1170
1210 RETURN
1220 REM UPDATE
1230 C = CHOICE
1240 N = N(C)
1250 FOR J = 1 TO N STEP 10
1260 REM DISPLAY
1270 GOSUB 1320
1280 REM SELECT
1290 GOSUB 1450
1300 NEXT J
1310 RETURN
1320 REM DISPLAY
1330 HOME
1340 PRINT L$
1350 LN = LEN (C$(C))
1360 PRINT TAB( 20 - LN / 2)C$(C)
1370 PRINT L$: PRINT
1380 R = 0:HT = 39:DOLL$ = "YES"
1390 M = J + L * 10
1400 FOR L = J TO J + 9
1410 IF L < = N THEN R = R + 1: INVERSE :
      PRINT CHR$( R + 64);: NORMAL : PRINT
      " ";X$(L,C); TAB( 26) "=";:NR = X(L,C):
      GOSUB 9000
1420 NEXT L
1430 VTAB 16: HTAB 1: PRINT L$

```



```

1440 RETURN
1450 REM SELECT
1460 VTAB 20: HTAB 1: PRINT SPC( 14)
1470 VTAB 22: HTAB 1: PRINT SPC( 39)
1480 VTAB 18: HTAB 1: PRINT "Change value
(Y/N) ? "; CHR$( 7);
1490 GET S$
1500 IF S$ = "N" OR S$ = "n" THEN 1640
1510 IF S$ < > "Y" AND S$ < > "y" THEN 1
480
1520 VTAB 20: HTAB 1: PRINT "Which letter
? "; CHR$( 7);
1530 GET S$
1540 A = ASC (S$)
1550 IF A > 90 THEN A = A - 32
1560 Q = A - 64
1570 IF Q < 1 OR Q > R THEN 1520
1580 VTAB (A - 60): HTAB 1: FLASH : PRINT
CHR$( A); CHR$( 7);: NORMAL
1590 VTAB 22: HTAB 1: INPUT "New value = ?
";S$
1600 X(J + Q - 1,C) = VAL (S$)
1610 VTAB (A - 60): HTAB 27: PRINT SPC( 1
3);:NR = X(J + Q - 1,C): GOSUB 9000
1620 VTAB (A - 60): HTAB 1: INVERSE : PRINT
CHR$( A);: NORMAL
1630 GOTO 1460
1640 RETURN
1650 REM DISPLAY
1660 HOME
1670 VTAB 12: HTAB 15: FLASH : PRINT "COMP
UTING": NORMAL
1680 REM COMPUTE
1690 GOSUB 1750
1700 REM CATEGORIES
1710 GOSUB 1920
1720 REM TOTALS
1730 GOSUB 2110
1740 RETURN
1750 REM COMPUTE
1760 REM TOTAL SPENDING IN EACH CATEGORY
1770 FOR I = 1 TO K
1780 T(I) = 0
1790 FOR J = 1 TO N(I)
1800 T(I) = T(I) + X(J,I)
1810 NEXT J,I
1820 REM GRAND TOTAL
1830 GT = 0
1840 FOR I = 1 TO K
1850 GT = GT + T(I)

```

```

1860 NEXT
1870 REM PERCENT
1880 FOR I = 1 TO K
1890 SPCT(I) = T(I) * 100 / GT
1900 NEXT
1910 RETURN
1920 REM CATEGORIES
1930 HOME
1940 PRINT L$
1950 PRINT TAB( 10)"MONTHLY EXPENDITURES
1960 PRINT L$
1970 PRINT TAB( 33)"Percent": PRINT TAB(
36)"of
1980 PRINT "Category"; TAB( 24)"Spending";
TAB( 34)"Total"
1990 ROW = 8
2000 FOR I = 1 TO K
2010 VTAB ROW: HTAB 1: PRINT C$(I);:DOLL$ =
"YES":HT = 31:DA = 2:NR = T(I): GOSUB
9000
2020 VTAB ROW:DOLL$ = "":HT = 38:DA = 1:NR
= SPCT(I): GOSUB 9000
2030 VTAB ROW: HTAB 39: PRINT "%"
2040 ROW = ROW + 1
2050 NEXT
2060 VTAB ROW + 1: HTAB 1: INVERSE : PRINT
"TOTAL";: NORMAL :DOLL$ = "YES":HT = 3
1:DA = 2:NR = GT: GOSUB 9000
2070 VTAB 22: HTAB 1: PRINT L$
2080 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
2090 GET S$
2100 RETURN
2110 REM TOTALS
2120 HOME
2130 PRINT L$
2140 PRINT TAB( 7)"GRAND TOTALS FOR THE M
ONTH
2150 PRINT L$
2160 VTAB 7: HTAB 1: PRINT "Total Income";
TAB( 16)"=";:HT = 32:NR = Y: GOSUB 90
00
2170 VTAB 9: HTAB 1: PRINT "Total Spending
"; TAB( 16)"=";:NR = GT: GOSUB 9000
2180 VTAB 11: HTAB 1: PRINT "Net Savings";
TAB( 16)"=";:NR = Y - GT: GOSUB 9000
2190 VTAB 17: HTAB 1: PRINT L$
2200 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
2210 GET S$

```

2220 RETURN
2230 REM DATA
2240 REM CATEGORIES
2250 DATA HOME OR APARTMENT, UTILITIES, T
TRANSPORTATION, FOOD, APPAREL, GIFTS &
GAIETY, EDUCATION, MISCELLANEOUS
2260 REM HOME OR APARTMENT
2270 DATA Mortgage payments,500
2280 DATA Property taxes,55
2290 DATA House insurance,25
2300 DATA House repairs,25
2310 DATA Rent,0
2320 REM UTILITIES
2330 DATA Oil,30
2340 DATA Phone,12
2350 DATA Electric,25
2360 DATA Water,13
2370 REM TRANSPORTATION
2380 DATA Car payments,95
2390 DATA Car insurance,20
2400 DATA Gasoline,30
2410 DATA Car repairs,25
2420 DATA Parking fees,25
2430 DATA Bus fare,20
2440 DATA Subway fare,20
2450 REM FOOD
2460 DATA Food,160
2470 DATA Drugstore items,10
2480 DATA Paper products,5
2490 DATA Pet food,10.98
2500 REM APPAREL
2510 DATA Dresses,75
2520 DATA Suits,50
2530 DATA Clothes for the kids,23
2540 DATA Shoes,30
2550 DATA Other,35
2560 REM ENTERTAINMENT & GIFTS
2570 DATA Movies,25
2580 DATA Dining out,125
2590 DATA Trips,35
2600 DATA Birthday presents,20
2610 DATA Church collection,40
2620 DATA United Fund,10
2630 REM EDUCATION
2640 DATA College tuition,200
2650 DATA College room & board,120
2660 DATA College expense money,35
2670 DATA School lunches,29
2680 DATA School books,7
2690 DATA Bus fare,10

```

2700 REM MISCELLANEOUS
2710 DATA Newspapers,20
2720 DATA Magazines,15
2730 DATA Computer hardware,150
2740 DATA Computer software,30
2750 DATA Hairdresser,35
2760 DATA Haircuts,12.50
2770 DATA Trash collection,22.80
2780 DATA Allowance for the kids,5
2790 DATA Other,0
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Car Trek

Whether you have a brand-new Porsche or a beat-up Volkswagen Beetle, "Car Trek" computes gas mileage for your road trips.

Most of us couldn't bear to live without our automobiles. But they're sometimes expensive to operate. Keeping track of your gas mileage is one way to find out how much your car is costing you. With "Car Trek," you can see exactly what mileage your car is getting.

You'll need to keep a few records for Car Trek's use. Jot down the odometer reading, the number of gallons of gasoline purchased, and the amount paid each time you fill up. A small notebook kept in the car will work fine. You also need the odometer reading at the start of your trip. Once you're home and back in front of the computer, you can use Car Trek to display some detailed figures of your car's expenses.

Behind the Wheel

Type in, save, and run Car Trek. Make sure you have the figures you wrote down when you put gas in your car. Then type in the beginning odometer reading, the number of gallons your car's tank holds (hit RETURN if you don't know), and then a figure representing how full your tank is (33 percent for one-third full, for instance).

The main display appears and you can begin entering the data you kept while on your road trip. Type in the amount (in gallons) and price of the gas you bought, as well as the odometer reading from each stop. You'll immediately see the mileage and the cost per gallon. From here you can continue (press C) to input data for another stop, redo (press R) the last entry, or tally (press T) the information.

When you ask the program to tally the figures, you have to enter a number showing how much gas is left in the tank. A summary will display, and you'll see how many miles you drove, the number of gallons your car consumed, and the average miles per gallon. At the bottom of the screen you'll see how much money you spent on gas, how many gallons were purchased, and an average price per gallon for the entire trip.

Car Trek

```

100 REM CAR TREK
110 REM TITLE
120 GOSUB 200
130 REM HEADING
140 GOSUB 250
150 REM START VALUES
160 GOSUB 390
170 REM TREK
180 GOSUB 810
190 END
200 REM TITLE
210 PRINT CHR$ (21); TEXT : HOME
220 VTAB 12: HTAB 16: PRINT "Car Trek
230 FOR PAUSE = 1 TO 2500: NEXT
240 RETURN
250 REM HEADING
260 HOME
270 PRINT "Car Trek computes gas mileage f
    or your
280 PRINT "automobile. Each time you fill
    'er up,
290 PRINT "please jot down the following:
300 PRINT
310 PRINT "1. Odometer reading
320 PRINT "2. Gallons purchased, and
330 PRINT "3. Amount paid.
340 PRINT
350 PRINT "Then I'll do the rest.
360 VTAB 23: HTAB 14: PRINT "Press any key
    ";
370 GET S$
380 RETURN
390 REM START VALUES
400 TMILES = 0: TGALS = 0: TCST = 0
410 STPS = 1
420 REM ODOMETER
430 GOSUB 520
440 REM GAS TANK CAPACITY
450 GOSUB 610
460 REM % FILLED
470 GOSUB 720
480 L$ = "": FOR I = 1 TO 39: L$ = L$ + "-":
    NEXT
490 REM DIGITS AFTER DECIMAL
500 DA = 2
510 RETURN
520 REM ODOMETER
530 HOME

```

```

540 PRINT "Please enter the odometer readi
ng at
550 PRINT "the start of your trek.
560 VTAB 4: HTAB 13: PRINT SPC( 15); CHR$
(7)
570 VTAB 4: HTAB 3: INPUT "Miles = ? ";S$
580 OHLD = VAL (S$)
590 IF OHLD < = 0 THEN 560
600 RETURN
610 REM GAS TANK CAPACITY
620 VTAB 7: HTAB 1: PRINT "Please enter th
e number of gallons that
630 PRINT "your gas tank holds. If you don
't know,"
640 PRINT "then simply hit the Return key
instead."
650 VTAB 11: HTAB 13: PRINT SPC( 15); CHR$
(7)
660 CAPACITY = 14
670 VTAB 11: HTAB 3: PRINT "Gallons = "; I
NVERSE : PRINT CAPACITY;: NORMAL
680 INPUT " ";G$
690 IF G$ < > "" THEN CAPACITY = VAL (G$
)
700 IF CAPACITY < = 0 THEN 650
710 RETURN
720 REM PERCENT FILLED
730 VTAB 14: HTAB 1: PRINT "Please enter t
he level of your gas
740 PRINT "gauge (% filled) at the start o
f your
750 PRINT "trek.
760 VTAB 18: HTAB 26: PRINT SPC( 15); CHR$
(7)
770 VTAB 18: HTAB 3: INPUT "Percent (0 to
100) = ? ";S$
780 PSTART = VAL (S$)
790 IF PSTART < 0 OR PSTART > 100 THEN 760
800 RETURN
810 REM TREK
820 REM DRAW BOX
830 GOSUB 990
840 REM INITIAL SLATE
850 GOSUB 1140
860 REM ENTER PIT-STOP DATA
870 GOSUB 1290
880 REM COMPUTE
890 GOSUB 1690
900 REM ENTER NEXT ACTION

```

```

910 GOSUB 1760
920 REM ADD TO TOTALS
930 IF R$ < > "R" THEN GOSUB 1880
940 REM PROCESS
950 IF R$ = "R" THEN VTAB 5: HTAB 3: PRINT
    "Let's try that again !": GOTO 870
960 IF R$ = "C" THEN 870
970 IF R$ = "T" THEN GOSUB 1960
980 RETURN
990 REM DRAW BOX
1000 HOME : INVERSE
1010 FOR I = 1 TO 39
1020 VTAB 1: HTAB I: PRINT CHR$ (32)
1030 VTAB 2: HTAB I: PRINT CHR$ (32)
1040 VTAB 3: HTAB I: PRINT CHR$ (32)
1050 VTAB 16: HTAB I: PRINT CHR$ (32)
1060 VTAB 24: HTAB I: PRINT CHR$ (32);
1070 NEXT
1080 FOR ROW = 1 TO 23
1090 VTAB ROW: HTAB 1: PRINT CHR$ (32)
1100 VTAB ROW: HTAB 39: PRINT CHR$ (32)
1110 NEXT ROW
1120 NORMAL
1130 RETURN
1140 REM INITIAL SLATE
1150 VTAB 2: HTAB 15: PRINT " CAR TREK "
1160 VTAB 5: HTAB 27: PRINT "Stop No."
1170 VTAB 6: HTAB 7: INVERSE : PRINT "PIT-
STOP DATA"; NORMAL
1180 VTAB 8: HTAB 13: PRINT "Gallons =
1190 VTAB 9: HTAB 13: PRINT "Payment =
1200 VTAB 10: HTAB 12: PRINT "Odometer =
1210 VTAB 12: HTAB 6: INVERSE : PRINT "CUR
RENT VALUES": NORMAL
1220 VTAB 14: HTAB 4: PRINT "Miles per Gal
lon =
1230 VTAB 15: HTAB 4: PRINT "Price per Gal
lon =
1240 VTAB 18: HTAB 6: PRINT "PRESS";; TAB(
27)"TO
1250 VTAB 20: HTAB 8: PRINT "C"; TAB( 22)"
Continue
1260 VTAB 21: HTAB 8: PRINT "R"; TAB( 22)"
Redo last entry
1270 VTAB 22: HTAB 8: PRINT "T"; TAB( 22)"
Tally totals
1280 RETURN
1290 REM ENTER DATA
1300 REM BLANK-OUT LINES
1310 GOSUB 1350

```



```

1320 REM DATA
1330 GOSUB 1430
1340 RETURN
1350 REM BLANK-OUT LINES
1360 FOR I = 8 TO 10
1370 VTAB I: HTAB 23: PRINT SPC( 14)
1380 NEXT
1390 FOR I = 14 TO 15
1400 VTAB I: HTAB 23: PRINT SPC( 14)
1410 NEXT
1420 RETURN
1430 REM DATA
1440 VTAB 5: HTAB 36: PRINT STPS
1450 REM GALLONS
1460 VTAB 8: HTAB 23: PRINT SPC( 15); CHR#
(7)
1470 VTAB 8: HTAB 22: INPUT " ";S#
1480 VTAB 8: HTAB 39: INVERSE : PRINT CHR#
(32): NORMAL
1490 GALS = VAL (S#)
1500 IF GALS < = 0 THEN VTAB 8: HTAB 23:
PRINT "No free ride !"; GOSUB 1640: GOTO
1460
1510 REM COST
1520 VTAB 9: HTAB 23: PRINT SPC( 15); CHR#
(7)
1530 VTAB 9: HTAB 22: INPUT " ";S#
1540 VTAB 9: HTAB 39: INVERSE : PRINT CHR#
(32): NORMAL
1550 CST = VAL (S#)
1560 IF CST < = 0 THEN VTAB 9: HTAB 23: PRINT
"I want more !"; GOSUB 1640: GOTO 152
0
1570 REM ODOMETER
1580 VTAB 10: HTAB 23: PRINT SPC( 15); CHR#
(7)
1590 VTAB 10: HTAB 22: INPUT " ";S#
1600 VTAB 10: HTAB 39: INVERSE : PRINT CHR#
(32): NORMAL
1610 ODOM = VAL (S#)
1620 IF ODOM < = 0 THEN VTAB 10: HTAB
23: PRINT "Go forward !"; GOSUB 1640:
GOTO 1580
1630 RETURN
1640 REM DELAY
1650 FOR PAUSE = 1 TO 250
1660 P = PEEK ( - 16336)
1670 NEXT PAUSE
1680 RETURN
1690 REM COMPUTE

```

```

1700 MILES = ODOM - OHLD
1710 MPG = MILES / GALS
1720 PPG = CST / GALS
1730 VTAB 14: HTAB 23: HT = 30: DOLL$ = "": NR = MPG: GOSUB 9000
1740 VTAB 15: HTAB 23: DOLL$ = "YES": NR = PPG: GOSUB 9000: DOLL$ = ""
1750 RETURN
1760 REM ENTER NEXT ACTION
1770 VTAB 18: HTAB 6: FLASH : PRINT "PRESS
"; CHR$ (7);: NORMAL
1780 P = PEEK ( - 16384)
1790 IF P < 128 THEN 1780
1800 R$ = CHR$ (P - 128)
1810 POKE - 16384,0
1820 A = ASC (R$)
1830 IF A > 90 THEN A = A - 32
1840 R$ = CHR$ (A)
1850 IF R$ < > "C" AND R$ < > "R" AND R$
< > "T" THEN 1770
1860 VTAB 18: HTAB 6: PRINT "PRESS"
1870 RETURN
1880 REM ADD TO TOTALS
1890 TMILES = TMILES + MILES
1900 TGALS = TGALS + GALS
1910 TCST = TCST + CST
1920 OHLD = ODOM
1930 STPS = STPS + 1
1940 VTAB 5: HTAB 3: PRINT SPC( 22)
1950 RETURN
1960 REM TOTALS
1970 REM FUEL GAUGE READING
1980 GOSUB 2040
1990 REM COMPUTATIONS
2000 GOSUB 2140
2010 REM DISPLAY
2020 GOSUB 2220
2030 RETURN
2040 REM FUEL GAUGE READING
2050 HOME
2060 PRINT "Please enter the level of your
gas
2070 PRINT "gauge (% filled) at the end of
your
2080 PRINT "trek.
2090 VTAB 5: HTAB 24: PRINT SPC( 15); CHR$
(7)
2100 VTAB 5: HTAB 1: INPUT "Percent (0 to
100) = ? "; S$
2110 PFINISH = VAL (S$)

```

```

2120 IF PFINISH < 0 OR PFINISH > 100 THEN
2090
2130 RETURN
2140 REM COMPUTATIONS
2150 REM PRICE PER GALLON PURCHASED
2160 PPG = TCST / TGALS
2170 REM TOTAL FUEL CONSUMED
2180 TF = TGALS + CAPACITY * (PSTART - PFIN
    ISH) / 100
2190 REM MILES PER GALLON
2200 MPG = TMILES / TF
2210 RETURN
2220 REM DISPLAY
2230 HOME
2240 PRINT L$
2250 PRINT TAB( 12)"CAR TREK SUMMARY
2260 PRINT L$
2270 VTAB 6: HTAB 1: INVERSE : PRINT "FUEL
    CONSUMPTION": NORMAL
2280 VTAB 8: HTAB 1: PRINT "Total miles dr
    iven"; TAB( 26)"="";HT = 39:NR = TMILE
    S: GOSUB 9000
2290 VTAB 9: HTAB 1: PRINT "Total gallons
    consumed"; TAB( 26)"="";NR = TF: GOSUB
    9000
2300 VTAB 10: HTAB 1: PRINT "Average miles
    per gallon"; TAB( 26)"="";NR = MPG: GOSUB
    9000
2310 VTAB 13: HTAB 1: INVERSE : PRINT "FUE
    L COST": NORMAL
2320 VTAB 15: HTAB 1: PRINT "Total dollars
    spent"; TAB( 26)"="";DOLL$ = "YES":NR
    = TCST: GOSUB 9000
2330 VTAB 16: HTAB 1: PRINT "Total gallons
    purchased"; TAB( 26)"="";DOLL$ = "":N
    R = TGALS: GOSUB 9000
2340 VTAB 17: HTAB 1: PRINT "Average price
    per gallon"; TAB( 26)"="";DOLL$ = "YE
    S":NR = PPG: GOSUB 9000
2350 VTAB 20: HTAB 1: PRINT L$
2360 VTAB 23: HTAB 14: PRINT "Press any ke
    y ";
2370 GET S$
2380 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
    00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
```

```

9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
    1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
    L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
    Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Weather Forecaster

Original Program Concept by George W. Miller

Everybody talks about the weather but nobody does anything about it. Although you may not be able to change the weather, with this program you can at least make a reasonably accurate local weather prediction.

It's easy to look at a threatening sky and predict that it will rain. But it's not always that easy to tell what the weather will be tomorrow, or even later that same day. The National Weather Service has been trying for years, and still it's impossible to predict with 100 percent accuracy what the weather will be.

You don't have the facilities and huge computers of the National Weather Service, but you do have a home computer that you can use to help forecast the weather.

The underlying principle of all weather-prediction computer models, which use hundreds of observations and scores of intricate equations, is simple. If we know what the current weather is and can correlate it with some past experience, then we can use our knowledge of what *was* to foretell what *may be*.

"Weather Forecaster" uses the same idea, but on a much smaller scale. The National Weather Service uses a network of reporting stations and satellites to gather its information. Since you don't have access to these, the best device available to you is a barometer. Barometric pressure, whether it's rising or falling, along with the wind direction, will allow you to make a fairly accurate local forecast.

To use this program, you'll need to know the current wind direction, barometric pressure, whether the barometer is rising or falling, and the rate of change of the barometer. The computer does the rest. You can buy an inexpensive barometer at most hardware stores, and a simple wind vane is easy to make. If you don't want to go to this trouble and expense, you can get the same information from the weather report on TV or from your local NOAA Weather Radio Station.

Local Conditions

Weather Forecaster will work fine as is. But you may want to add additional information about local weather conditions so

that you can customize the program to your area. This information does not change the operation of the program; it is just a convenience.

Program lines 2540–2660 contain the weather norms. Each DATA statement corresponds to a month of the year and contains five numbers. The first three are temperatures: the normal high, normal low, and normal average for the month, in degrees Fahrenheit. The last two numbers are normal monthly rainfall and snowfall, in inches.

Contact a local TV station, newspaper, or National Weather Service reporting station to get the values for your area. Or write NOAA, *National Environmental Satellite, Data, and Information Service, National Climatic Data Center, Federal Building, Asheville, NC 28801*, and request a copy of the Local Climatological Data Annual Summary for your area. Almanacs sometimes include this information as well. *The Weather Almanac*, edited by James A. Ruffner and Frank E. Blair, published by Avon Books, is available in most libraries. As you enter the program, change the DATA statements accordingly. For instance, in the month of July, your area may have a normal high of 86, a normal low of 59, an average temperature of 73, and normal rainfall of 2.1 inches (with no snowfall, of course). To reflect this local condition, you would change line 2610 to read:

DATA July,86,59,73,2.1,0.0

It's easier if you locate this information before you begin typing in the program; you can then change the DATA statements as you enter them.

After you've saved and run Weather Forecaster, you'll see a short menu. Pressing a key sends you to one of three routines.

Forecast the weather. This is the heart of the program. Pressing the 1 key changes the display and shows a prompt asking for the wind direction (the direction the wind is blowing from). Refer to the compass on the screen and enter the number representing the current wind direction. As an example, enter 6, for winds from the SW.

Next, enter the present barometric pressure, say, 30.29 (the program will round it up to 30.3). Then type in a number from 1 to 5; each represents a barometric trend, from steady to falling fast. Enter 1 for steady.

As soon as you enter this last number, the forecast appears. Your values are displayed (just to make sure you entered them correctly) and the forecast shows on the screen. You should see this line displayed: *Forecast: Continued fair with little or no change in temperature.* Press any key to return to the main menu.

Display monthly weather norms. By pressing the 2 key, you can see what the normal temperature and precipitation values are for your area. (Remember that you have to change the DATA statements in lines 2540–2660 to reflect actual conditions in your area.) Entering another number, from 1 to 12, displays a month's norms. If you had changed the norms as in the example described above, when you pressed the 7 key for the month of July, you would see 86 for a high, 59 for a low, 73 for an average temperature, 2.1 inches of rainfall, and no snowfall. Hitting any key switches you back to the main menu.

Exit. Press the 3 key when you're through.

Fine-Tuning

Changes in barometric pressure and wind direction can imply different forecasts for different parts of the country. You may want to fine-tune the program even more to fit your area. In doing this, note that:

- Lines 1700–1900 actually do the forecasting, with three variables used:
 - B = Barometric pressure
 - W = Wind direction
 - T = Barometric trend
- Barometric trend, in turn, takes on any one of five values:
 - 1 = Steady
 - 2 = Rising slowly
 - 3 = Rising rapidly
 - 4 = Falling slowly
 - 5 = Falling rapidly
- The forecasts that appear on the screen are in DATA statements in lines 2710–2880. The array variables F\$(n) in lines 1700–1900 call these forecasts to the screen.

You'll have to do some research at the local library or contact the National Weather Service if you want to modify this part of the program.

Weather Forecaster

```

100 REM WEATHER FORECASTER
110 REM INITIALIZE
120 GOSUB 180
130 REM SELECT FROM MENU
140 GOSUB 700
150 ON PICK GOSUB 920,2210
160 IF PICK < > 3 THEN 140
170 END
180 REM INITIALIZE
190 REM TITLE
200 GOSUB 260
210 REM INSTRUCTIONS
220 GOSUB 310
230 REM KEY VALUES
240 GOSUB 450
250 RETURN
260 REM TITLE
270 PRINT CHR$(21): TEXT : HOME
280 VTAB 12: HTAB 11: PRINT "Weather Forec
    aster
290 FOR PAUSE = 1 TO 2000: NEXT
300 RETURN
310 REM INSTRUCTIONS
320 HOME
330 PRINT " This program helps you to mak
    e
340 PRINT "accurate, short-range weather f
    orecasts.";
350 PRINT
360 PRINT " You'll need to know the curre
    nt wind
370 PRINT "direction, barometric pressure,
    and
380 PRINT "whether the barometer is rising
    or
390 PRINT "falling.
400 PRINT
410 PRINT " I'll do the rest.
420 VTAB 23: HTAB 14: PRINT "Press any key
    ";
430 GET S$
440 RETURN
450 REM KEY VALUES
460 L$ = "": FOR I = 1 TO 39: L$ = L$ + "=":
    NEXT
470 REM NUMBER OF FORECASTS
480 DATA 17
490 READ K

```



```

500 DIM M$(12),MW(12,5),F$(K)
510 REM MONTHLY WEATHER NORMS
520 FOR I = 1 TO 12
530 READ M$(I)
540 FOR J = 1 TO 5
550 READ MW(I,J)
560 NEXT J,I
570 REM WIND DIRECTION
580 FOR I = 1 TO 8
590 READ D$(I)
600 NEXT
610 REM BAROMETER TREND
620 FOR I = 1 TO 5
630 READ BT$(I)
640 NEXT
650 REM FORECASTS
660 FOR I = 1 TO K
670 READ F$(I)
680 NEXT
690 RETURN
700 REM MENU
710 REM HEADING
720 GOSUB 760
730 REM ACTION
740 GOSUB 830
750 RETURN
760 REM HEADING
770 HOME
780 VTAB 10: HTAB 3: FLASH : PRINT "PRESS"
;: INVERSE : HTAB 19: PRINT "TO"
790 VTAB 12: HTAB 5: PRINT "1";: NORMAL : HTAB
10: PRINT "Forecast the weather
800 VTAB 14: HTAB 5: INVERSE : PRINT "2";:
NORMAL : HTAB 10: PRINT "Display mont
hly weather norms
810 VTAB 16: HTAB 5: INVERSE : PRINT "3";:
NORMAL : HTAB 10: PRINT "Exit
820 RETURN
830 REM ACTION
840 PRINT CHR$(7);
850 P = PEEK (- 16384)
860 IF P < 128 THEN 850
870 R$ = CHR$(P - 128)
880 PICK = VAL (R$)
890 POKE - 16368,0
900 IF PICK < 1 OR PICK > 3 THEN 840
910 RETURN
920 REM FORECAST
930 REM ENTER WIND DIRECTION
940 GOSUB 1000

```

```

950 REM ENTER BAROMETER INFO
960 GOSUB 1340
970 REM PREDICT
980 GOSUB 1640
990 RETURN
1000 REM WIND
1010 REM VANE
1020 GOSUB 1060
1030 REM ENTRY
1040 GOSUB 1230
1050 RETURN
1060 REM VANE
1070 HOME
1080 PRINT L$
1090 PRINT TAB( 13)"WIND DIRECTION
1100 PRINT L$
1110 VTAB 11: HTAB 35: PRINT D$(1): VTAB 7
: HTAB 31: PRINT D$(2)
1120 VTAB 4: HTAB 26: PRINT D$(3): VTAB 7:
HTAB 20: PRINT D$(4)
1130 VTAB 11: HTAB 17: PRINT D$(5): VTAB 1
5: HTAB 20: PRINT D$(6)
1140 VTAB 18: HTAB 26: PRINT D$(7): VTAB 1
5: HTAB 31: PRINT D$(8)
1150 INVERSE : FOR ROW = 6 TO 16
1160 VTAB ROW: HTAB 26: PRINT CHR$ (32)
1170 NEXT
1180 FOR COL = 19 TO 33
1190 VTAB 11: HTAB COL: PRINT CHR$ (32)
1200 NEXT
1210 NORMAL
1220 RETURN
1230 REM ENTRY
1240 FOR I = 1 TO 8
1250 VTAB 2 * I + 2: HTAB 5: INVERSE : PRINT
I;: NORMAL : PRINT " ";D$(I)
1260 NEXT
1270 VTAB 20: HTAB 1: PRINT "Please enter
the direction that the
1280 PRINT "wind is blowing from.
1290 VTAB 23: HTAB 1: PRINT "Number = ? ";
CHR$ (7);
1300 GET S$
1310 W = VAL (S$)
1320 IF W < 1 OR W > 8 THEN 1290
1330 RETURN
1340 REM BAROMETER INFO
1350 REM PRESENT READING
1360 GOSUB 1400

```

```

1370 REM TREND
1380 GOSUB 1530
1390 RETURN
1400 REM PRESENT READING
1410 HOME
1420 PRINT L$
1430 PRINT TAB( 10)"BAROMETRIC PRESSURE
1440 PRINT L$
1450 VTAB 5: HTAB 1: PRINT "What is the pr
     esent barometric
1460 VTAB 6: HTAB 12: PRINT SPC( 15); CHR$
     (7)
1470 VTAB 6: HTAB 1: INPUT "pressure ? ";P
     $
1480 B = VAL (P$)
1490 VTAB 23: HTAB 8: PRINT SPC( 32)
1500 IF B < = 0 THEN VTAB 23: HTAB 8: PRINT
     "You must be on the moon !": GOTO 1460

1510 IF B > 50 THEN VTAB 23: HTAB 10: PRINT
     "You must be kidding !": GOTO 1460
1520 RETURN
1530 REM TREND
1540 VTAB 8: HTAB 1: INVERSE : PRINT "PRES
     ENT BAROMETRIC TREND": NORMAL
1550 FOR I = 1 TO 5
1560 VTAB 2 * I + 8: HTAB 1: INVERSE : PRINT
     I;: NORMAL : PRINT " ";BT$(I)
1570 NEXT
1580 PRINT
1590 VTAB 20: HTAB 1: PRINT "Number = ? ";
     CHR$ (7);
1600 GET S$
1610 T = VAL (S$)
1620 IF T < 1 OR T > 5 THEN 1590
1630 RETURN
1640 REM PREDICT
1650 REM COMPUTE
1660 GOSUB 1700
1670 REM DISPLAY
1680 GOSUB 1910
1690 RETURN
1700 REM COMPUTE
1710 HOME
1720 F$ = "Fair, little change in temperatu
     re for the next day or two
1730 IF B > = 30.2 AND T = 4 AND (W > =
     4 AND W < = 6) THEN F$ = F$(1)
1740 IF B > = 30.2 AND T = 1 AND (W > =
     4 AND W < = 6) THEN F$ = F$(2)

```

```

1750 IF (B > = 30.1 AND B < 30.2) AND T =
    1 AND (W > = 4 AND W < = 6) THEN F$ =
    F$(3)
1760 IF B > = 30.1 AND T = 3 AND (W > =
    4 AND W < = 6) THEN F$ = F$(4)
1770 IF (B > = 30.1 AND B < 30.2) AND T =
    4 AND (W > = 4 AND W < = 6) THEN F$ =
    F$(5)
1780 IF B > = 30.1 AND T = 5 AND (W > =
    4 AND W < = 6) THEN F$ = F$(6)
1790 IF B > = 30.1 AND T = 4 AND W = 7 THEN
    F$ = F$(7)
1800 IF B > = 30.1 AND T = 5 AND W = 7 THEN
    F$ = F$(8)
1810 IF B > = 30.1 AND T = 4 AND (W = 2 OR
    W = 1 OR W = 8) THEN F$ = F$(9)
1820 IF B > = 30.1 AND T = 5 AND W = 8 THEN
    F$ = F$(10)
1830 IF B > = 30.1 AND T = 5 AND (W = 1 OR
    W = 2) THEN F$ = F$(11)
1840 IF B < = 29.8 AND T = 5 AND (W > =
    1 AND W < = 3) THEN F$ = F$(12)
1850 IF B < = 29.8 AND T = 5 AND (W = 8 OR
    W = 7) THEN F$ = F$(13)
1860 IF B < = 29.8 AND T = 3 THEN F$ = F$
    (14)
1870 IF B < 30.1 AND T = 4 AND (W = 2 OR W
    = 1 OR W = 8) THEN F$ = F$(15)
1880 IF (B > 29.8 AND B < 30.1) AND T = 5 AND
    (W = 2 OR W = 1 OR W = 8) THEN F$ = F$
    (16)
1890 IF B < = 30.1 AND T = 2 AND (W = 7 OR
    W = 6) THEN F$ = F$(17)
1900 RETURN
1910 REM DISPLAY
1920 PRINT L$
1930 PRINT TAB( 12)"WEATHER FORECAST
1940 PRINT L$
1950 VTAB 5: HTAB 1: PRINT "Current barome
    tric
1960 PRINT "pressure"; TAB( 21)"= ";B
1970 PRINT
1980 PRINT "Barometric trend"; TAB( 21)"=
    ";BT$(T)
1990 PRINT
2000 PRINT "Wind direction"; TAB( 21)"= ";
    D$(W)
2010 VTAB 12: HTAB 1: INVERSE : PRINT "FOR
    ECAST:"; NORMAL
2020 LF$ = F$:RG$ = ""

```

```

2030 LN = LEN (F$)
2040 IF LN > 39 THEN GOSUB 2110
2050 VTAB 14: HTAB 1: PRINT LF$
2060 PRINT RG$
2070 VTAB 19: HTAB 1: PRINT L$
2080 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
2090 GET S$
2100 RETURN
2110 REM DIVIDE LINE
2120 P = 0: W$ = LEFT$ (F$,39)
2130 FOR I = 39 TO 2 STEP - 1
2140 C$ = RIGHT$ (W$,1)
2150 IF C$ = CHR$ (32) THEN P = I: I = 2
2160 W$ = LEFT$ (W$, I - 1)
2170 NEXT
2180 LF$ = LEFT$ (F$,P - 1)
2190 RG$ = RIGHT$ (F$,LN - P)
2200 RETURN
2210 REM MONTHLY NORMS
2220 REM ENTER MONTH
2230 GOSUB 2270
2240 REM DISPLAY
2250 GOSUB 2380
2260 RETURN
2270 REM ENTER MONTH
2280 HOME
2290 VTAB 4: HTAB 1: PRINT "Please enter t
he month you'd like.
2300 FOR I = 1 TO 6
2310 VTAB I * 2 + 5: HTAB 5: INVERSE : PRINT
I;; HTAB 23: PRINT I + 6;; NORMAL : HTAB
7: PRINT M$(I);; HTAB 26: PRINT M$(I +
6)
2320 NEXT
2330 VTAB 20: HTAB 16: PRINT SPC( 15); CHR$
(7)
2340 VTAB 20: HTAB 5: INPUT "Number = ? ";
N$
2350 N = INT ( VAL (N$))
2360 IF N < 1 OR N > 12 THEN 2330
2370 RETURN
2380 REM DISPLAY
2390 HOME
2400 PRINT L$
2410 PRINT TAB( 20 - LEN (M$(N)) / 2);M$
(N)
2420 PRINT L$
2430 VTAB 5: HTAB 1: INVERSE : PRINT "TEMP
ERATURE": NORMAL

```

```

2440  VTAB 7: HTAB 1: PRINT "Normal high"; TAB(
      17)"= ";MW(N,1);" degrees F.
2450  VTAB 8: HTAB 1: PRINT "Normal low"; TAB(
      17)"= ";MW(N,2);" degrees F.
2460  VTAB 9: HTAB 1: PRINT "Normal average
      "; TAB( 17)"= ";MW(N,3);" degrees F.
2470  VTAB 12: HTAB 1: INVERSE : PRINT "PRE
      CIPITATION": NORMAL
2480  VTAB 14: HTAB 1: PRINT "Normal rainfa
      ll = ";MW(N,4);" inches
2490  VTAB 15: HTAB 1: PRINT "Normal snowfa
      ll = ";MW(N,5);" inches
2500  VTAB 19: HTAB 1: PRINT L$
2510  VTAB 23: HTAB 14: PRINT "Press any ke
      y ";
2520  GET S$
2530  RETURN
2540  REM MONTHLY WEATHER NORMS (Normal Hi
      gh, Low, & Average; Rain & Snow)
2550  DATA January,38,23,30,2.8,9.1
2560  DATA February,41,24,32,2.7,9.6
2570  DATA March,51,31,41,3.2,6.5
2580  DATA April,64,42,53,3.0,0.3
2590  DATA May,75,52,63,3.6,0.0
2600  DATA June,83,61,72,3.6,0.0
2610  DATA July,87,65,76,3.6,0.0
2620  DATA August,85,63,74,3.8,0.0
2630  DATA September,78,56,67,3.2,0.0
2640  DATA October,67,45,56,2.8,0.1
2650  DATA November,53,35,44,2.7,2.1
2660  DATA December,40,25,33,2.9,7.7
2670  REM WIND DIRECTION
2680  DATA E, NE, N, NW, W, SW, S, SE
2690  REM BAROMETER TREND
2700  DATA Steady, Rising slowly, Rising f
      ast, Falling slowly, Falling fast
2710  REM FORECASTS
2720  DATA "Fair and warmer for the next 4
      8 hours
2730  DATA "Continued fair with little or
      no change in temperature
2740  DATA "Fair, little change in tempera
      ture for the next day or two
2750  DATA "Fair today, rainy and warmer w
      ithin 48 hours
2760  DATA "Warmer, rain within 24 to 36 h
      ours
2770  DATA "Warmer, rain within 18 to 24 h
      ours

```

2780 DATA "Rain within 24 hours
2790 DATA "Windy, rain within 12 to 24 h
ours
2800 DATA "Rain in 12 to 18 hours
2810 DATA "Windy, rain within 12 hours
2820 DATA "In summer, rain within 12 to 2
4 hrs. In winter, rain or snow, & wind
y
2830 DATA "Heavy rain in summer. In winte
r, heavy snow followed by a cold wave
2840 DATA "Severe storm warning. Windy, w
ith rain in summer & snow in winter
2850 DATA "Clearing and colder
2860 DATA "Rain for the next day or two
2870 DATA "Rain with high winds; clearing
and cooler within 24 hours
2880 DATA "Clearing within a few hours; f
air for the next several days

Calorie Cop

Original Program Concept by Gerald P. Graham

"Calorie Cop" determines your caloric expenditure for over 100 different activities. Whether you're just sitting around or exercising, this program will count the calories you burn during a day. Several menus and easy input make it simple to use.

You may know approximately how many calories you're taking in each day, but in order to know if you're gaining or losing weight, you also have to know how many calories you burn up. "Calorie Cop" can tell you, with just a few keypresses. Over 100 activities are listed in Calorie Cop, enough to help you develop your physical fitness program or weight reduction plan.

Evaluating Exercise

Type in and save Calorie Cop. Run it, and the computer first asks your weight. This is needed to correctly figure your caloric expenditure. The heavier you are, the more calories you use up in any given activity. Of course, the heavier you are, the more you may want to lose if you're planning a reduction program. It all evens out in the end. As an example, type in 160 for your weight.

The remainder of the program is a series of activity menus. Each menu (except for the last) includes ten activities labeled from A to J. The first menu's activities include exercises that range from archery to bicycling slowly on level ground. The activities are arranged in alphabetical order.

To use these menus, find the appropriate activity, press the Y key to indicate you want to make a selection, then hit the activity's letter key. If you want to find out how many calories are expended by bicycling slowly, press Y, then J. Calorie Cop reveals that you'll burn 4.8 calories per minute (assuming you entered 160 as your weight).

Respond to the next prompt with the number of minutes for this activity. For example, 15 minutes of bicycling expends 72 calories. The computer also keeps a running total of how many calories you've burned up so far. It's a nice way to evaluate an entire day's exercise.

If you don't see your activity on the first menu screen, just hit the N key and the next menu appears. Remember that the activities are listed in alphabetical order. Keep moving through the menus until you find the activity you want.

Just as with any other diet/exercise plan, do not use this program except under the advice and consent of your physician.

Calorie Cop

```

100 REM CALORIE COP
110 REM INITIALIZE
120 REM TITLE
130 GOSUB 230
140 REM INSTRUCTIONS
150 GOSUB 280
160 REM READ DATA
170 GOSUB 490
180 REM DISPLAY & SELECT
190 GOSUB 560
200 REM TOTALS
210 GOSUB 1130
220 END
230 REM TITLE
240 PRINT CHR$(21): TEXT : HOME
250 VTAB 12: HTAB 15: PRINT "CALORIE COP
260 FOR PAUSE = 1 TO 2000: NEXT
270 RETURN
280 REM INSTRUCTIONS
290 HOME
300 REM NUMBER OF EXERCISES
310 DATA 125
320 READ N
330 DIM E$(N),C(N)
340 PRINT "I'M CALORIE COP. I COMPUTE HOW
    MUCH
350 PRINT "ENERGY YOU'LL BURN UP IN PERFOR
    MING A
360 PRINT "HOST OF ACTIVITIES.
370 PRINT
380 PRINT "TO DO THIS, HOWEVER, I NEED TO
    ASK YOU"
390 PRINT "A PERSONAL QUESTION.
400 VTAB 8: HTAB 33: PRINT SPC(15); CHR$(
    7)
410 VTAB 8: HTAB 1: INPUT "NAMELY, HOW MUC
    H DO YOU WEIGH ? ";S$
420 WEIGHT = VAL(S$)
430 IF WEIGHT < 1 OR WEIGHT > 500 THEN VTAB
    23: HTAB 16: PRINT "COME ON !": GOTO 4
    00

```

```

440 TL = 0
450 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT I
460 REM DIGITS AFTER DECIMAL
470 DA = 2
480 RETURN
490 REM READ
500 HOME
510 VTAB 12: HTAB 16: FLASH : PRINT "READI
    NG": NORMAL
520 FOR I = 1 TO N
530 READ E$(I),C(I)
540 NEXT I
550 RETURN
560 REM DISPLAY & SELECT
570 FOR J = 1 TO N STEP 10
580 REM DISPLAY
590 GOSUB 640
600 REM SELECT
610 GOSUB 750
620 NEXT J
630 RETURN
640 REM DISPLAY
650 HOME
660 PRINT L$
670 PRINT TAB( 14)"ACTIVITY MENU
680 PRINT L$: PRINT
690 R = 0
700 FOR L = J TO J + 9
710 IF L < = N THEN R = R + 1: INVERSE : PRINT
    CHR$( R + 64);: NORMAL : PRINT " ";E$
    (L)
720 NEXT L
730 VTAB 16: HTAB 1: PRINT L$
740 RETURN
750 REM SELECT
760 VTAB 18: HTAB 1: PRINT "SELECT (Y/N) ?
    "; CHR$( 7);
770 GET S$
780 IF S$ = "N" OR S$ = "n" THEN 900
790 IF S$ < > "Y" AND S$ < > "y" THEN 76
    0
800 VTAB 18: HTAB 20: PRINT "WHICH LETTER
    ? "; CHR$( 7);
810 GET S$
820 A = ASC (S$)
830 IF A > 90 THEN A = A - 32
840 Q = A - 64
850 IF Q < 1 OR Q > R THEN 800
860 REM ENTER LENGTH OF EXERCISE

```

```

870 GOSUB 910
880 REM DISPLAY SAME BATCH AGAIN
890 GOSUB 640: GOTO 760
900 RETURN
910 REM LENGTH
920 HOME : PRINT L$
930 Q = J + Q - 1
940 LN = LEN (E$(Q))
950 PRINT TAB( 20 - LN / 2)E$(Q)
960 PRINT L$
970 VTAB 20: HTAB 1: PRINT L$
980 CAL = C(Q) * WEIGHT
990 VTAB 5: HTAB 1: PRINT "CALORIES BURNED
    UP PER MINUTE = ";:HT = 39:NR = CAL: GOSUB
    9000
1000 VTAB 7: HTAB 32: PRINT SPC( 15); CHR$(
    7)
1010 VTAB 7: HTAB 1: INPUT "HOW MANY MINUT
    ES DO YOU WANT ? ";S$
1020 TIME = VAL (S$)
1030 IF TIME < 0 THEN VTAB 23: HTAB 16: PRINT
    "COME ON !": GOTO 1000
1040 REM CALORIE COUNT
1050 CURRENT = TIME * CAL
1060 TL = TL + CURRENT
1070 VTAB 10: HTAB 1: INVERSE : PRINT "CAL
    ORIE COUNT": NORMAL
1080 VTAB 12: HTAB 1: PRINT "THIS EXERCISE
    "; TAB( 15)"=";:HT = 27:NR = CURRENT: GOSUB
    9000
1090 VTAB 13: HTAB 1: PRINT "CUMULATIVE"; TAB(
    15)"=";:NR = TL: GOSUB 9000
1100 VTAB 23: HTAB 14: PRINT "PRESS ANY KE
    Y ";
1110 GET S$
1120 RETURN
1130 REM TOTALS
1140 HOME
1150 PRINT "TOTAL CALORIES USED = ";:HT =
    30:NR = TL: GOSUB 9000
1160 RETURN
1170 REM DATA
1180 DATA ARCHERY, .034
1190 DATA BADMINTON - MODERATE, .039
1200 DATA BADMINTON - VIGOROUS, .065
1210 DATA BASEBALL, .031
1220 DATA BASEBALL - PITCH & CATCH, .04
1230 DATA BASKETBALL - MODERATE, .047
1240 DATA BASKETBALL - VIGOROUS, .066

```

1250	DATA	BED MAKING, .031
1260	DATA	BICYCLING - DOWNHILL, .018
1270	DATA	BICYCLING - LEVEL GROUND (SLOW) , .03
1280	DATA	BICYCLING - MODERATE, .050
1290	DATA	BICYCLING - UPHILL (FAST), .072
1300	DATA	BOXING - IN RING, .101
1310	DATA	BOXING - SPARRING, .063
1320	DATA	BOWLING, .028
1330	DATA	CANOEING, .029
1340	DATA	CONVERSING, .011
1350	DATA	COOKING, .013
1360	DATA	DANCING - SLOW, .029
1370	DATA	DANCING - MODERATE, .045
1380	DATA	DANCING - FAST, .064
1390	DATA	DRESSING & UNDESSING, .030
1400	DATA	DRIVING A CAR, .019
1410	DATA	DUSTING, .010
1420	DATA	EATING, .011
1430	DATA	EXERCISES - ABDOMINAL, .020
1440	DATA	EXERCISES - BALANCING, .016
1450	DATA	EXERCISES - JUMPING, .043
1460	DATA	EXERCISES - BENDING, .023
1470	DATA	FENCING - MODERATE, .033
1480	DATA	FENCING - VIGOROUS, .057
1490	DATA	FIELD HOCKEY, .063
1500	DATA	FIELD HOCKEY - GOALIE, .030
1510	DATA	FISHING, .016
1520	DATA	FOOTBALL - BACKS & ENDS, .050
1530	DATA	FOOTBALL - LINEMAN, .040
1540	DATA	GARDENING, .030
1550	DATA	GOLF - CROWDED & WALKING, .030
1560	DATA	GOLF - UNCROWDED & WALKING, .03
5		
1570	DATA	GYMNASTICS - LIGHT, .030
1580	DATA	GYMNASTICS - HEAVY, .056
1590	DATA	HANDBALL, .063
1600	DATA	HIKING, .042
1610	DATA	HILL & STAIR CLIMBING, .060
1620	DATA	HORSEBACK RIDING - WALK, .019
1630	DATA	HORSEBACK RIDING - TROT, .046
1640	DATA	HORSEBACK RIDING - GALLOP, .067
1650	DATA	IRONING, .018
1660	DATA	JUDO, .087
1670	DATA	JUMPING ROPE, .087
1680	DATA	KARATE, .087
1690	DATA	LACROSSE, .063
1700	DATA	LACROSSE - GOALIE, .030
1710	DATA	LISTENING TO RADIO, .010
1720	DATA	MOTOR BOATING, .016

1730	DATA	MOUNTAIN CLIMBING, .086
1740	DATA	PAINTING - INSIDE, .015
1750	DATA	PAINTING - OUTSIDE, .035
1760	DATA	PLAYING CARDS, .011
1770	DATA	PLAYING DRUMS, .030
1780	DATA	PLAYING HORN, .013
1790	DATA	PLAYING PIANO, .018
1800	DATA	RACQUETBALL, .063
1810	DATA	RESTING - LYING DOWN, .008
1820	DATA	RESTING - SITTING, .009
1830	DATA	ROWING - SLOW, .036
1840	DATA	ROWING - VIGOROUS, .118
1850	DATA	RUNNING - 11 MINUTES PER MILE, .071
1860	DATA	RUNNING - 10 MINUTES PER MILE, .078
1870	DATA	RUNNING - 9 MINUTES PER MILE, .085
1880	DATA	RUNNING - 8 MINUTES PER MILE, .092
1890	DATA	RUNNING - 7 MINUTES PER MILE, .100
1900	DATA	RUNNING - 6 MINUTES PER MILE, .110
1910	DATA	RUNNING - 5 MINUTES PER MILE, .130
1920	DATA	SAILING, .020
1930	DATA	SCRUBBING, .032
1940	DATA	SEWING OR KNITTING, .010
1950	DATA	SHOPPING, .028
1960	DATA	SHOWERING, .034
1970	DATA	SINGING - STANDING, .017
1980	DATA	SITTING - QUIETLY, .010
1990	DATA	SITTING - WRITING, .013
2000	DATA	SKATING - MODERATE, .036
2010	DATA	SKATING - VIGOROUS, .064
2020	DATA	SKIING - DOWNHILL, .059
2030	DATA	SKIING - LEVEL GROUND (SLOW), .054
2040	DATA	SKIING - LEVEL GROUND (FAST), .078
2050	DATA	SLEEPING, .007
2060	DATA	SOCCER, .063
2070	DATA	SOCCER - GOALIE, .030
2080	DATA	SQUASH, .070
2090	DATA	STANDING, .011
2100	DATA	STATIONARY RUNNING, .078
2110	DATA	STUDYING, .014
2120	DATA	SWIM - CRAWL (30 YDS/MIN.), .05

2130 DATA SWIM - CRAWL (40 YDS/MIN.), .07
 1
 2140 DATA SWIM - BACKSTROKE (30 YDS/MIN.)
 , .035
 2150 DATA SWIM - BACKSTROKE (40 YDS/MIN.)
 , .055
 2160 DATA SWIM - BREASTSTROKE (30 YDS/MIN
 .), .048
 2170 DATA SWIM - BREASTSTROKE (40 YDS/MIN
 .), .064
 2180 DATA SWIM - BUTTERFLY, .078
 2190 DATA TABLE TENNIS - MODERATE, .026
 2200 DATA TABLE TENNIS - VIGOROUS, .040
 2210 DATA TELEPHONING, .011
 2220 DATA TENNIS - SINGLES - MODERATE, .0
 46
 2230 DATA TENNIS - SINGLES - VIGOROUS, .0
 65
 2240 DATA TENNIS - DOUBLES - MODERATE, .0
 38
 2250 DATA TENNIS - DOUBLES - VIGOROUS, .0
 46
 2260 DATA TYPING, .015
 2270 DATA VOLLEYBALL - BEGINNERS - MODERA
 TE, .020
 2280 DATA VOLLEYBALL - BEGINNERS - VIGORO
 US, .036
 2290 DATA VOLLEYBALL - SKILLED - MODERATE
 , .040
 2300 DATA VOLLEYBALL - SKILLED - VIGOROUS
 , .065
 2310 DATA WALKING - 2 MILES PER HOUR, .02
 2
 2320 DATA WALKING - 3 MILES PER HOUR, .03
 0
 2330 DATA WALKING - 4 MILES PER HOUR, .03
 9
 2340 DATA WALKING - 5 MILES PER HOUR, .06
 4
 2350 DATA WASHING DISHES, .015
 2360 DATA WASHING HANDS & FACE, .020
 2370 DATA WATCHING TV, .010
 2380 DATA WATER SKIING, .053
 2390 DATA WEIGHT LIFTING - ARMS, .050
 2400 DATA WEIGHT LIFTING - LEGS, .060
 2410 DATA WEIGHT LIFTING - BODY, .065
 2420 DATA WRESTLING, .091
 9000 REM "PRINT USING" SUBROUTINE
 9010 ZR\$ = "":ZS\$ = "":ZD\$ = "":ZZ\$ = "0000
 00000"

```

9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Utility-Bill Audit

Original Program Concept by Larry L. Bihlmeyer

As utility rates continue to climb, it's a good idea to check your monthly bills for accuracy. "Utility-Bill Audit" verifies your electric, gas, and water bills, and even lets you split them between several people in your household.

Sometimes utility bills seem unreasonable even if they are calculated accurately. But if they're wrong, you need to know. Most of us don't know how our bills are calculated or even keep track of the meter readings. With this program, however, you can verify the accuracy of your bills quickly. And you can split the cost between several people in your household. If you're interested in cutting costs on your utilities, you can even monitor your daily electric and gas consumption.

Rates

Type in and save "Utility-Bill Audit." Before you run it, however, you need to have some information about your own utility bills in order to customize the program.

Most utility rates depend on the amount of electricity, gas, or water you use during a month. In other words, the cost per unit depends on your total consumption. Some companies charge less for each unit when you use a lot. Others charge more per unit, the more you use.

You need to know this rate breakdown before you use Utility-Bill Audit. Look at your most recent bills to find this information. If it's not there, contact your utility company to get a rate schedule.

After you run the program, select which bill you would like to calculate by pressing the keys indicated in the menu. Now you need to insert your own data in place of what is used in the program.

Look at lines 580-790. These are the lines that contain sample utility-bill rates. For example, the sample electric rate is 6.622 cents for each kilowatt hour of electricity up to 800 kilowatt hours. Line 600 contains this rate information in the form:

```
DATA 0,800,6.622 :REM START, END, CENTS
```


The first number is the start of the range, the second the end of the range, and the third the rate per kilowatt hour. Line 610 has the information for the second part of the rate schedule. It's organized the same way. Notice that the rate is 7.439 cents per kilowatt hour for between 801 and 9999 kilowatt hours. Line 620 indicates that the rate is 7.439 cents for each kilowatt hour over 9999 kilowatt hours. The tax rate and maximum tax are in lines 610 and 620. In the default arrangement, there is an 8 percent tax, with a maximum tax of \$4.00. The other utilities (gas and water) are listed in the following lines.

You need to replace these numbers with your own rates. Your schedule will probably be different from this one, both in actual rates and in the cutoff limits. If your electric rate for the first 500 kilowatt hours is 8.45 cents, for example, you'd replace line 600 with:

DATA 0,500,8.45 :REM START, END, CENTS

If your tax rate was only 5 percent, and the maximum tax limit was \$2.00, you'd replace lines 640 and 650 with:

640 DATA 5

650 DATA 2

(If you don't have a maximum utility tax in your area, use 0 for the second DATA statement and delete line 2030 in the program.) Go through and make the necessary changes to the program.

Notice that in the default arrangement you pay the same rate for both gas and water, no matter how much you use. Your situation may be different. Once you have these changes inserted in the program, save it again under a different name. This is the version you should use when you audit your bills. *Always keep the original working version for reference.*

Readings

Run the program and you'll see a main menu, which allows you to choose between calculating your electric, gas, or water bills. (When you're finished with the program, you can hit the 4 key to exit.)

Let's check an electric bill. We'll use the default values for this example. Press the 1 key to select electricity, and the screen display changes. The rate schedule is shown, as well as another menu at the bottom of the screen. To change a price per unit or tax rate, press C, then make the alterations. Since

you've already changed the basic rate schedule information, you'll probably use this option only when your utility costs or taxes increase. Hitting the R key returns you to the main menu and pressing the T key tallies your bill.

Press T. Enter both your previous and present meter readings. You can find these on your latest bill, or if you're monitoring your daily electric use, you can read the meter yourself. As an example, enter 3000 for the previous reading and 4000 for the present reading. Answer the next prompt, which asks for the number of people who are sharing this bill. Enter 1 if you're paying for the entire bill yourself. Press the RETURN key and the totals appear. You've used 1000 kilowatt hours of electricity and paid 7.439 cents per kilowatt hour, so your basic charge is \$74.39. Eight percent of that (for the tax) is \$5.95, more than the \$4.00 maximum, so you pay the latter. The total bill is \$78.39.

Auditing a gas or water bill is just as easy. As usual, the hard part is paying the bills.

Utility-Bill Audit

```

100 REM UTILITY BILL AUDIT
110 REM INITIALIZE
120 GOSUB 170
130 REM SELECT FROM MENU
140 GOSUB 900
150 IF C < > 4 THEN GOSUB 1020: GOTO 140
160 END
170 REM INITIALIZE
180 REM TITLE
190 GOSUB 250
200 REM MESSAGE
210 GOSUB 300
220 REM KEY VALUES
230 GOSUB 420
240 RETURN
250 REM TITLE
260 PRINT CHR$(21): TEXT : HOME
270 VTAB 13: HTAB 11: PRINT "Utility Bill
    Audit
280 FOR PAUSE = 1 TO 2000: NEXT
290 RETURN
300 REM MESSAGE
310 HOME
320 PRINT "This program helps you:
330 PRINT
340 PRINT " -- Check your Electric, Gas, a
    nd Water
    
```

```

350 PRINT "      bills for accuracy.
360 PRINT
370 PRINT " -- Split costs among the people living"
380 PRINT "      in your household.
390 VTAB 23: HTAB 14: PRINT "Press any key
    ";
400 GET S$
410 RETURN
420 REM KEY VALUES
430 DATA 3
440 READ K
450 DATA Electric Bill, Gas Bill, Water Bill
460 FOR I = 1 TO K
470 READ U$(I)
480 NEXT
490 REM UNITS OF MEASUREMENT
500 DATA Kilowatt-hours, Cubic feet of gas, Gallons of water (000)
510 FOR I = 1 TO K
520 READ M$(I)
530 NEXT
540 REM CUTOFF LIMITS & CHARGES
550 GOSUB 580
560 L$ = "": FOR I = 1 TO 39: L$ = L$ + "=":
    NEXT
570 RETURN
580 REM CUTOFF LIMITS & CHARGES
590 REM KILOWATTHOURS
600 DATA 0,800,6.622 : REM START,END,
    CENTS
610 DATA 801,9999,7.439 : REM START,END,
    CENTS
620 DATA 7.439 : REM CENTS
630 REM TAX RATE (% & MAX CHARGE)
640 DATA 8
650 DATA 4
660 REM CUBIC FEET OF GAS
670 DATA 0,1000,0.6416
680 DATA 1001,9999,0.6416
690 DATA 0.6416
700 REM TAX RATE
710 DATA 8
720 DATA 4
730 REM THOUSANDS OF GALLONS OF WATER
740 DATA 0,1000,70
750 DATA 1001,9999,70
760 DATA 70
770 REM TAX RATE

```

```

780 DATA 8
790 DATA 4
800 FOR I = 1 TO K
810 FOR J = 1 TO 2
820 READ LOWER(I,J),UPPER(I,J)
830 READ PRICE(I,J)
840 NEXT J
850 READ PRICE(I,3)
860 READ TAX(I)
870 READ MTAX(I)
880 NEXT I
890 RETURN
900 REM MENU
910 HOME
920 VTAB 6: HTAB 9: PRINT "Would you like
to
930 VTAB 8: HTAB 10: INVERSE : PRINT "1";:
NORMAL : PRINT " Compute your electri
c bill
940 VTAB 10: HTAB 10: INVERSE : PRINT "2";
: NORMAL : PRINT " Compute your gas bi
ll
950 VTAB 12: HTAB 10: INVERSE : PRINT "3";
: NORMAL : PRINT " Compute your water
bill
960 VTAB 14: HTAB 10: INVERSE : PRINT "4";
: NORMAL : PRINT " Exit
970 VTAB 16: HTAB 9: PRINT "Choice = ? "; CHR$
(7);
980 GET S$
990 C = VAL (S$)
1000 IF C < 1 OR C > 4 THEN 970
1010 RETURN
1020 REM COMPUTE
1030 REM HEADING
1040 GOSUB 1150
1050 REM CHARGES
1060 GOSUB 1230
1070 REM TAXES
1080 GOSUB 1370
1090 REM ENTER NEXT ACTION
1100 GOSUB 1430
1110 IF R$ = "C" THEN GOSUB 1580
1120 IF R$ = "T" THEN GOSUB 1700
1130 IF R$ < > "R" THEN GOTO 1040
1140 RETURN
1150 REM HEADING
1160 L1 = LEN (U$(C))
1170 HOME
1180 PRINT L$

```

```

1190 PRINT TAB( 20 - L1 / 2)U$(C)
1200 PRINT L$
1210 VTAB 5: HTAB 1: PRINT M$(C); TAB( 26)
    "Price Per Unit
1220 RETURN
1230 REM CHARGES
1240 PRINT
1250 FOR J = 1 TO 2
1260 VTAB J + 6: HTAB 1: INVERSE : PRINT J
    ;: NORMAL :HT = 6:DA = 0:DOLL$ = "":NR
    = LOWER(C,J): GOSUB 9000
1270 VTAB J + 6: HTAB 9: PRINT "to"
1280 VTAB J + 6:HT = 16:NR = UPPER(C,J): GOSUB
    9000
1290 VTAB J + 6:HT = 33:DA = 4:NR = PRICE(
    C,J): GOSUB 9000
1300 NEXT
1310 VTAB 9: HTAB 1: INVERSE : PRINT "3";:
    NORMAL : PRINT " More than";:HT = 16:
    DA = 0:NR = UPPER(C,2): GOSUB 9000
1320 VTAB 9:HT = 33:DA = 4:NR = PRICE(C,3)
    : GOSUB 9000
1330 FOR I = 7 TO 9
1340 VTAB I: HTAB 35: PRINT "cents"
1350 NEXT
1360 RETURN
1370 REM TAXES
1380 VTAB 11: HTAB 1: INVERSE : PRINT "4";
    : NORMAL : PRINT " Tax rate"; TAB( 23)
    "=:;:HT = 31:DA = 2::NR = TAX(C): GOSUB
    9000
1390 VTAB 11: HTAB 33: PRINT "%"
1400 VTAB 12: HTAB 1: INVERSE : PRINT "5";
    : NORMAL : PRINT " Maximum monthly tax
    =";:DOLL$ = "YES":NR = MTAX(C): GOSUB
    9000
1410 VTAB 14: HTAB 1: PRINT L$
1420 RETURN
1430 REM ENTER NEXT ACTION
1440 VTAB 15: HTAB 6: FLASH : PRINT "PRESS
    ";;: NORMAL : PRINT TAB( 27)"TO
1450 VTAB 17: HTAB 8: PRINT "C"; TAB( 18)"
    Change a price or tax
1460 VTAB 18: HTAB 8: PRINT "R"; TAB( 18)"
    Return to main menu
1470 VTAB 19: HTAB 8: PRINT "T"; TAB( 18)"
    Tally bill"; CHR$( 7)
1480 PRINT L$
1490 P = PEEK ( - 16384)
1500 IF P < 128 THEN 1490

```

```

1510 R$ = CHR$ (P - 128)
1520 POKE - 16368,0
1530 IF R$ = "c" THEN R$ = "C"
1540 IF R$ = "r" THEN R$ = "R"
1550 IF R$ = "t" THEN R$ = "T"
1560 IF R$ < > "C" AND R$ < > "R" AND R$
    < > "T" THEN 1490
1570 RETURN
1580 REM CHANGE
1590 VTAB 15: HTAB 6: PRINT "PRESS"
1600 VTAB 21: HTAB 1: PRINT "Number (1 to
    5) = ? "; CHR$ (7);
1610 GET S$
1620 N = VAL (S$)
1630 IF N < 1 OR N > 5 THEN 1600
1640 PRINT N
1650 VTAB 23: HTAB 1: INPUT "New Value = ?
    ";V$
1660 IF N < = 3 THEN PRICE(C,N) = VAL (V
    $)
1670 IF N = 4 THEN TAX(C) = VAL (V$)
1680 IF N = 5 THEN MTAX(C) = VAL (V$)
1690 RETURN
1700 REM TALLY BILL
1710 REM METER READINGS
1720 GOSUB 1780
1730 REM COMPUTE BILL
1740 GOSUB 1970
1750 REM DISPLAY BILL
1760 GOSUB 2060
1770 RETURN
1780 REM METER READINGS
1790 HOME
1800 PRINT L$
1810 PRINT TAB( 20 - L1 / 2)U$(C)
1820 PRINT L$
1830 PRINT "Please enter your meter readin
    gs.
1840 VTAB 7: HTAB 14: PRINT SPC( 15)
1850 VTAB 8: HTAB 14: PRINT SPC( 15); CHR$
    (7)
1860 VTAB 7: HTAB 1: INPUT "Previous = ? "
    ;M1$
1870 VTAB 8: HTAB 1: INPUT " Present = ? "
    ;M2$
1880 UNITS = VAL (M2$) - VAL (M1$)
1890 IF UNITS < 0 THEN VTAB 22: HTAB 5: PRINT
    "Your consumption is negative !": GOTO
    1840

```

```

1900 VTAB 22: HTAB 5: PRINT SPC( 35)
1910 VTAB 10: HTAB 1: PRINT "How many peop
    le will share the
1920 VTAB 11: HTAB 8: PRINT SPC( 15); CHR$
    (7)
1930 VTAB 11: HTAB 1: INPUT "bill ? ";S$
1940 N = INT ( VAL (S$))
1950 IF N < 1 THEN 1920
1960 RETURN
1970 REM BILL
1980 IF UNITS > = LOWER(C,1) AND UNITS <
    = UPPER(C,1) THEN Z = 1
1990 IF UNITS > = LOWER(C,2) AND UNITS <
    = UPPER(C,2) THEN Z = 2
2000 IF UNITS > UPPER(C,2) THEN Z = 3
2010 BASE = UNITS * PRICE(C,Z) / 100
2020 TX = BASE * TAX(C) / 100
2030 IF TX > MTAX(C) THEN TX = MTAX(C)
2040 BILL = BASE + TX
2050 RETURN
2060 REM DISPLAY
2070 HOME
2080 PRINT L$
2090 PRINT TAB( 20 - L1 / 2)U$(C)
2100 PRINT L$
2110 VTAB 5: HTAB 1: PRINT "QUANTITY: ";UN
    ITS;" ";M$(C)
2120 VTAB 9: HTAB 1: INVERSE : PRINT "UTIL
    ITY BILL";: HTAB 16: PRINT "HOUSEHOLD"
    ;: HTAB 30: PRINT "PER PERSON": NORMAL

2130 VTAB 11: HTAB 1: PRINT "Basic charge"
    ;:DOLL$ = "YES":DA = 2:HT = 24:NR = BA
    SE: GOSUB 9000
2140 VTAB 11:HT = 39:NR = BASE / N: GOSUB
    9000
2150 VTAB 12: HTAB 1: PRINT "Tax";:HT = 24
    :NR = TX: GOSUB 9000
2160 VTAB 12:HT = 39:NR = TX / N: GOSUB 90
    00
2170 VTAB 14: HTAB 1: PRINT "Total";:HT =
    24:NR = BILL: GOSUB 9000
2180 VTAB 14:HT = 39:NR = BILL / N: GOSUB
    9000
2190 VTAB 22: HTAB 1: PRINT L$
2200 VTAB 23: HTAB 14: PRINT "Press any ke
    y ";
2210 GET S$
2220 RETURN
9000 REM "PRINT USING" SUBROUTINE

```

```

9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
    00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
    1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
    L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
    Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```


Paycheck Analysis

Original Program Concept by Larry L. Bihlmeyer

Check your paycheck for accuracy or project your future take-home pay with this short program.

It's not impossible. Your employer may have made a mistake and calculated your paycheck incorrectly. If you're like most people, it's probably happened to you at least once. With this program, however, you can verify that the figures are right. Or if you're an employer, you can use this to insure that your employees are getting all they're entitled to.

Tax Schedules

Before you can use this program, however, you'll probably want to customize it a bit. As it's listed, the program calculates withholding for a single person. If you're in another classification, such as married filing jointly or married filing separately, you need to change some of the program lines. And the rates are likely to change from year to year.

Nevertheless, all you need is a copy of the tax rate schedules, which you can obtain from the Internal Revenue Service. Ask for Schedule X. You will also need information on your state taxes. Then insert the new data for federal taxes in lines 1290-1460. If you look at these program lines, you'll see that taxes are computed using the function $A + B * (Y - C)$. The variable Y is your annual income, or weekly pay times 52. A, B, and C are from the tax tables. According to line 1340, for example, your tax would be $\$241 + 0.15 * (Y - \$4,400)$ if Y is between \$4,400 and \$8,500. In this case, then:

A = 241
B = 0.15
C = 4400

To change the rates, place the new information in the lines using the same format.

State taxes are computed in lines 1210-1280. As listed, they calculate the withholding for the state of Virginia. The same formula as the federal withholding is used. You can replace these lines with ones that match your own state's tax rates (assuming your state has an income tax).

Of course, you'll have to alter both the federal and state tax rates if they change in the coming years.

Deductions and Salary

As soon as you've made these customizations, you can check your paycheck. Just type in, save, and run this program. The display shows such things as your base salary, overtime pay rate and hours, number of exemptions, insurance, United Fund, stock plan, and credit union deductions. Number 9, *F.I.C.A.*, stands for the Federal Insurance Contributions Act, or Social Security. Presently, 6.7 percent of your paycheck is deducted for this. If Congress changes the rate, make sure you alter line 420 in the program.

To enter your own figures in any of the categories, simply press the Y key, then the number key that represents the item. Type in the new figure and press the RETURN key. The display reappears, showing your change.

When you've entered your paycheck's information, the computer tallies your take-home pay as soon as you press the N key to signal no further changes are needed. Your gross pay, total taxes, and other deductions are listed, as well as the amount you'll actually take home. If it doesn't match what is on your check, perhaps you should talk to your employer.

Projections

You can also use the program to consider "what if?" situations. What if you get that raise you've been promised? What if you get ten hours of overtime this week? Simply return to the first display and change some of the figures. You can recalculate your paycheck as many times as you like.

Note that the program computes only weekly checks. If you're paid on a different schedule, say twice each month or even monthly, you'll need to modify the program accordingly. It's not difficult. Just find all the places in the program where variables are multiplied or divided by 52, and change them to 26 for biweekly or 12 for monthly paychecks. You'll also need the proper schedules from the IRS to change the withholding routine in lines 1290-1460.

Paycheck Analysis

```
100 REM REM PAYCHECK ANALYSIS
110 REM INITIALIZE
120 GOSUB 220
```

```

130 REM MENU
140 GOSUB 530
150 REM COMPUTE TAKE-HOME PAY
160 GOSUB 890
170 VTAB 21: HTAB 10: PRINT "Compute again
    (Y/N) ? "; CHR$(7);
180 GET S$
190 IF S$ = "Y" OR S$ = "y" THEN 140
200 IF S$ < > "N" AND S$ < > "n" THEN 17
    0
210 END
220 REM INITIALIZE
230 REM TITLE
240 GOSUB 280
250 REM VALUES
260 GOSUB 330
270 RETURN
280 REM TITLE
290 PRINT CHR$(21): TEXT : HOME
300 VTAB 12: HTAB 14: PRINT "Take-Home Pay

310 FOR PAUSE = 1 TO 2000: NEXT
320 RETURN
330 REM VALUES
340 DIM NM$(20),X(20)
350 REM NUMBER OF ITEMS
360 DATA 9
370 READ N
380 REM ITEM NAMES
390 DATA Base Salary $, Hourly Overtime R
    ate $, No. of Overtime Hours, No. of F
    ederal Exemptions
400 DATA Insurance $, United Fund $, Stoc
    k Plan $, Credit Union Deduction $,F.I
    .C.A.
410 REM CORRESPONDING VALUES
420 DATA 200,10,0,1,15,10,50,50,6.7
430 FOR I = 1 TO N
440 READ NM$(I)
450 NEXT
460 FOR I = 1 TO N
470 READ X(I)
480 NEXT
490 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT
500 REM DIGITS AFTER DECIMAL
510 DA = 2
520 RETURN
530 REM MENU
540 REM DISPLAY

```

```

550 GOSUB 640
560 REM UPDATE
570 VTAB 22: HTAB 1: PRINT SPC( 40)
580 VTAB 19: HTAB 1: PRINT "Would you like
    to change a
590 VTAB 20: HTAB 1: PRINT "value (Y/N) ?
    "; CHR$( 7);
600 GET S$
610 IF S$ = "Y" OR S$ = "y" THEN GOSUB 76
    0: GOTO 570
620 IF S$ < > "N" AND S$ < > "n" THEN 59
    0
630 RETURN
640 REM MENU
650 HOME
660 PRINT L$
670 PRINT TAB( 10)"WEEKLY TAKE-HOME PAY
680 PRINT L$
690 VTAB 5: HTAB 1: INVERSE : PRINT "PAY B
    ASE"; HTAB 29: PRINT "TAX FIGURES"
700 FOR I = 1 TO N
710 VTAB I + 7: HTAB 1: INVERSE : PRINT I;
    : NORMAL : PRINT " ";NM$(I);:DOLL$ = "
    ":HT = 38:NR = X(I): GOSUB 9000
720 NEXT
730 VTAB 16: HTAB 39: PRINT "% "
740 VTAB 18: HTAB 1: PRINT L$
750 RETURN
760 REM UPDATE
770 VTAB 22: HTAB 1: PRINT "Number = ? "; CHR$(
    7);
780 GET S$
790 V = VAL (S$)
800 IF V < 1 OR V > N THEN 770
810 VTAB V + 7: HTAB 1: FLASH : PRINT V: NORMAL

820 VTAB 22: HTAB 1: PRINT SPC( 10)
830 VTAB 22: HTAB 1: INPUT "New Value = ?
    ";S$
840 X(V) = VAL (S$)
850 VTAB V + 7: HTAB 1: INVERSE : PRINT V:
    NORMAL
860 VTAB V + 7: HTAB 28: PRINT SPC( 11)
870 VTAB V + 7:NR = X(V): GOSUB 9000
880 RETURN
890 REM TAKE-HOME PAY (GROSS)
900 GS = X(1) + X(2) * X(3)
910 REM MAXIMUM WEEKLY SALARY FOR FICA
920 MX = 37800 / 52
930 FICA = GS * X(9) / 100

```

```

940 IF GS > MX THEN FICA = MX * X(9) / 100
950 REM STATE OF VIRGINIA TAXES (TAXABLE
    INCOME PER WEEK)
960 Y = GS - (600 / 52) * X(4)
970 GOSUB 1210
980 REM FEDERAL TAXES (TAXABLE INCOME PER
    WEEK)
990 Y = GS - (1000 / 52) * X(4)
1000 GOSUB 1290
1010 REM OTHER DEDUCTIONS
1020 OTHER = X(5) + X(6) + X(7) + X(8)
1030 REM TAKE-HOME PAY
1040 PAY = GS - FICA - STAX - FTAX - OTHER
1050 REM DISPLAY
1060 GOSUB 1080
1070 RETURN
1080 REM DISPLAY
1090 HOME
1100 PRINT L$
1110 PRINT TAB( 14)"TAKE-HOME PAY
1120 PRINT L$
1130 VTAB 6: HTAB 1: INVERSE : PRINT "GROS
    S PAY";: NORMAL : PRINT TAB( 19)"=";:
    DOLL$ = "YES":HT = 34:NR = GS: GOSUB 9
    000
1140 VTAB 8: HTAB 2: PRINT "F.I.C.A."; TAB(
    19)"=";:NR = FICA: GOSUB 9000
1150 VTAB 9: HTAB 2: PRINT "Federal Taxes"
    ; TAB( 19)"=";:NR = FTAX: GOSUB 9000
1160 VTAB 10: HTAB 2: PRINT "State Taxes";
    TAB( 19)"=";:NR = STAX: GOSUB 9000
1170 VTAB 12: HTAB 2: PRINT "Other Deducti
    ons"; TAB( 19)"=";:NR = OTHER: GOSUB 9
    000
1180 VTAB 15: HTAB 1: INVERSE : PRINT "TAK
    E-HOME PAY";: NORMAL : PRINT TAB( 19)
    "=";:NR = PAY: GOSUB 9000
1190 VTAB 18: HTAB 1: PRINT L$
1200 RETURN
1210 REM STATE TAXES
1220 Y = Y * 52
1230 IF Y < = 3000 THEN A = 0:B = .02:C =
    0
1240 IF Y > 3000 AND Y < = 5000 THEN A =
    60:B = .03:C = 3000
1250 IF Y > 5000 AND Y < = 12000 THEN A =
    120:B = .05:C = 5000
1260 IF Y > 12000 THEN A = 470:B = .0575:C
    = 12000
1270 STAX = (A + B * (Y - C)) / 52

```

```

1280 RETURN
1290 REM FEDERAL TAXES
1300 Y = Y * 52
1310 IF Y < = 2300 THEN A = 0:B = 0:C = 0
1320 IF Y > 2300 AND Y < = 3400 THEN A =
      0:B = .11:C = 2300
1330 IF Y > 3400 AND Y < = 4400 THEN A =
      121:B = .13:C = 3400
1340 IF Y > 4400 AND Y < = 8500 THEN A =
      251:B = .15:C = 4400
1350 IF Y > 8500 AND Y < = 10800 THEN A =
      866:B = .17:C = 8500
1360 IF Y > 10800 AND Y < = 12900 THEN A =
      1257:B = .19:C = 10800
1370 IF Y > 12900 AND Y < = 15000 THEN A =
      1656:B = .21:C = 12900
1380 IF Y > 15000 AND Y < = 18200 THEN A =
      2097:B = .24:C = 15000
1390 IF Y > 18200 AND Y < = 23500 THEN A =
      2865:B = .28:C = 18200
1400 IF Y > 23500 AND Y < = 28800 THEN A =
      4349:B = .32:C = 23500
1410 IF Y > 28800 AND Y < = 34100 THEN A =
      6045:B = .36:C = 28800
1420 IF Y > 34100 AND Y < = 41500 THEN A =
      7953:B = .4:C = 34100
1430 IF Y > 41500 AND Y < = 55300 THEN A =
      10913:B = .45:C = 41500
1440 IF Y > 55300 THEN A = 17123:B = .5:C =
      55300
1450 FTAX = (A + B * (Y - C)) / 52
1460 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
      00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
      1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
      L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
      Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
      "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
      ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

3 Number Crunchers

五
五
五
五
五

五
五
五
五
五

Number Crunchers

Manipulating numbers is one of the things a computer does best. Your computer can perform calculations accurately and quickly. In fact, not too many years ago, you would have needed a huge, expensive mainframe computer to do the same computations your Apple is capable of.

This chapter provides you with a host of handy algorithms designed to tap some of the tremendous number-crunching capability of your Apple computer.

The first four programs, "Sort," "Random Number Generator," "Numerical Integration," and "Derivative," are useful in many home, school, and business activities. Employ these alone or as subroutines in larger programs of your own.

The last four programs involve matrix math. Matrices are rectangular arrays of numbers or variables. They enable you to represent unwieldy arithmetic relationships or large groups of data in an easy-to-understand fashion. Because of their great utility, matrices commonly appear in everyday computer programs.

These eight number crunchers are:

- **Sort.** Arrange a group of numbers in high-to-low or low-to-high sequence.
- **Random Number Generator.** Pluck values from a normal distribution.
- **Numerical Integration.** Compute the area under a curve.
- **Derivative.** Compute a variable's rate of change.
- **Matrix Addition and Subtraction.** Perform simple arithmetic calculations on groups of data.
- **Matrix Multiplication.** Calculate the product of two groups of data.
- **Matrix Inversion.** Solve sets of simultaneous equations.
- **MatMan.** Painlessly evaluate lengthy matrix expressions.

A Note on Numbers

Anytime you work with measurement and numbers, you should keep in mind the possible sources of error. Today's home computers are generally very accurate, but are on occasion subject to rounding errors. Use common sense when

interpreting the results obtained from these programs. If the answer to a derivative calculation is 1000.0001, you should consider rounding this to 1000 if you use it in another calculation. The difference is most likely insignificant, since it is smaller than the possible error in the calculation.

Sort

Sorting numbers by hand can be tiresome. Fortunately, your computer can do it for you, and in a matter of moments. All you have to do is use this program, entering the numbers you want sorted. The program gives you the option of sorting into either low-to-high or high-to-low order.

As you sort numbers, you probably search through the list for the highest value (if you're sorting in descending order), cross it off, and then copy it onto another piece of paper. With only a few numbers that might not be too time-consuming. But if you have dozens, or hundreds, of numbers, they can be sorted more efficiently by using your computer.

Although your computer handles the task in much the same way, it's faster. So fast, in fact, that it takes less time to sort than it does to input the numbers.

Made to Order

Type in this short program and save it. Run it, and you can choose the sort sequence. Press the 1 key if you want to sort the numbers from low to high, the 2 key if you want them organized from high to low. Then enter the number of values you want to sort, followed by the values themselves. It's that simple.

Try it out. You'll find many uses for this simple sort, from organizing test scores to zip codes.

Sort

```
100 REM SORT
110 REM INITIALIZE
120 GOSUB 200
130 REM ENTER NUMBERS
140 GOSUB 540
150 REM SORT
160 GOSUB 800
170 REM DISPLAY
180 GOSUB 1070
190 END
200 REM INITIALIZE
210 REM TITLE
220 GOSUB 260
230 REM HEADING
```

Number Crunchers

```
240 GOSUB 360
250 RETURN
260 REM TITLE
270 PRINT CHR$(21): TEXT : HOME
280 DATA ORTS,RTSO,TSOR,SORT
290 FOR I = 1 TO 4
300 READ W$
310 VTAB 12: HTAB 18: PRINT W$
320 FOR T = 1 TO 500: NEXT T
330 NEXT I
340 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT I
350 RETURN
360 REM HEADING
370 REM MAXIMUM NUMBER OF VALUES
380 DATA 300
390 READ NX
400 DIM X(NX)
410 REM GREETING
420 HOME
430 PRINT "This program sorts numbers. Would you
440 PRINT "like them in
450 VTAB (4): HTAB (11): PRINT "1. Low to
    High, or
460 HTAB (11): PRINT "2. High to Low
470 PRINT
480 HTAB (9): PRINT "sequence (1 or 2) ? "
    ;
490 PRINT CHR$(7);
500 GET S$
510 CHOICE = INT ( VAL (S$))
520 IF CHOICE < > 1 AND CHOICE < > 2 THEN
    490
530 RETURN
540 REM ENTER NUMBERS
550 REM TOTAL COUNT
560 GOSUB 600
570 REM VALUES
580 GOSUB 700
590 RETURN
600 REM TOTAL
610 HOME
620 PRINT "How many values do you want to
630 VTAB 2: HTAB 8: PRINT SPC(15)
640 VTAB 2: HTAB 1: PRINT CHR$(7);: INPUT
    "sort ? ";N$
650 N = INT ( VAL (N$))
660 VTAB 15: HTAB 1: PRINT SPC(39)
```

```
670 IF N < 2 THEN VTAB 15: HTAB 5: PRINT
    "I need at least two numbers !": GOTO
    630
680 IF N > NX THEN VTAB 15: HTAB 9: PRINT
    "Only ";NX;" are allowed !": GOTO 630
690 RETURN
700 REM VALUES
710 HOME
720 PRINT "Please enter your data.
730 FOR I = 1 TO N
740 VTAB 3: HTAB 14: PRINT SPC( 15)
750 VTAB 3: HTAB 1: PRINT "No. ( ";I; TAB(
    10) ")= ";
760 INPUT V$
770 X(I) = VAL (V$)
780 NEXT
790 RETURN
800 REM SORT
810 HOME
820 FLASH
830 VTAB 13: HTAB 15: PRINT "SORTING";
840 NORMAL
850 ON CHOICE GOSUB 870,1000
860 RETURN
870 REM LOW TO HIGH
880 SWAP$ = "NO"
890 FOR J = 1 TO N - 1
900 IF X(J) > X(J + 1) THEN GOSUB 940
910 NEXT J
920 IF SWAP$ = "YES" THEN 880
930 RETURN
940 REM SWAP
950 HOLD = X(J)
960 X(J) = X(J + 1)
970 X(J + 1) = HOLD
980 SWAP$ = "YES"
990 RETURN
1000 REM HIGH TO LOW
1010 SWAP$ = "NO"
1020 FOR J = 1 TO N - 1
1030 IF X(J) < X(J + 1) THEN GOSUB 940
1040 NEXT J
1050 IF SWAP$ = "YES" THEN 1010
1060 RETURN
1070 REM DISPLAY
1080 T$ = "Low to High"
1090 IF CHOICE = 2 THEN T$ = "High to Low"

1100 T$ = "ORDER: " + T$
```

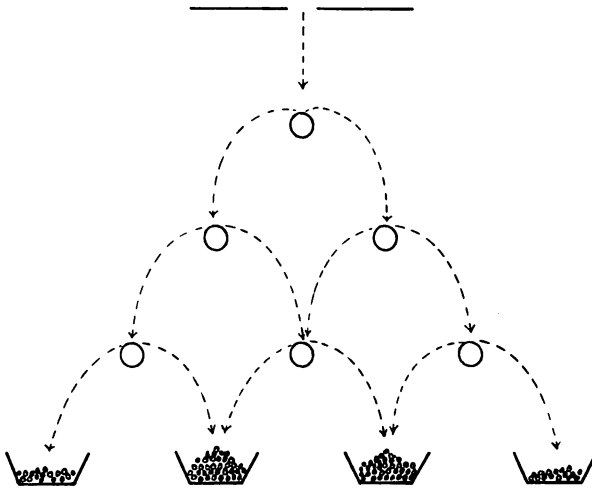
```
1110 FOR L = 0 TO INT ((N - 1) / 10)
1120 GOSUB 1150
1130 NEXT L
1140 RETURN
1150 REM 10 VALUES AT A TIME
1160 HOME
1170 PRINT L$
1180 PRINT TAB( 14)"SORTED VALUES"
1190 PRINT TAB( 11)T$
1200 PRINT L$: PRINT
1210 FOR J = 1 TO 10
1220 Q = J + L * 10
1230 IF Q < = N THEN PRINT TAB( 3)"No.
    ( ";Q; TAB( 13)")= ";X(Q)
1240 NEXT J
1250 VTAB 22: HTAB 1: PRINT L$
1260 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
1270 GET S$
1280 RETURN
```

Random Number Generator

Generating random numbers within a distribution is simple when you use this program.

A pellet-dropping machine is a board with pegs in the middle and cups at the bottom. The board is propped up and pellets are dropped through a hole at the top center of the board. The pellets hit the top peg and bounce left or right, eventually working their way into the cups below. The spread of pellets among the bottom bins resembles the normal distribution, especially when many pegs and cups are used. The machine might look like Figure 1.

Figure 1. Pellet Machine

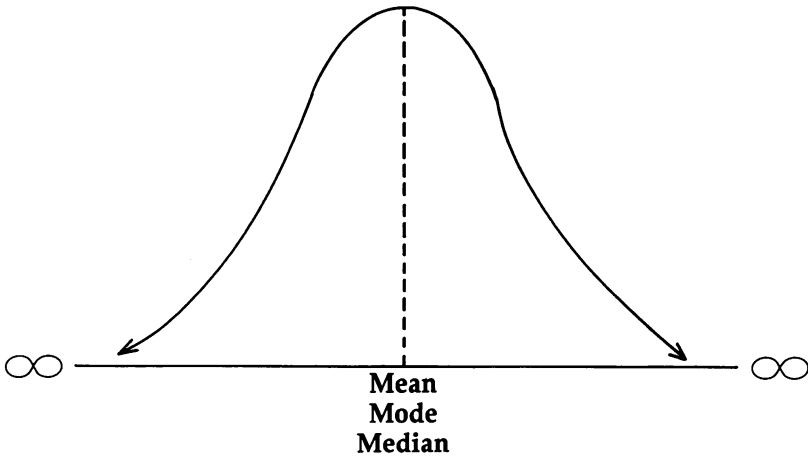


This program does somewhat the same thing. It generates numbers within a *normal distribution*, a popular and powerful tool of statistics. A number of commonplace physical and social phenomena approximate it, such as IQ scores, the amount of liquid in 16-ounce bottles, and people's weights. The normal distribution is used in computer simulations of all sorts, in everything from business planning to games to scientific investigations. If you're hoping to use a normal distribution in

your own programs, this routine will be extremely helpful. First, however, a few elementary definitions.

The *normal curve* is bell shaped and extends from minus to plus infinity, with mean, mode, and median all equal, as Figure 2 shows.

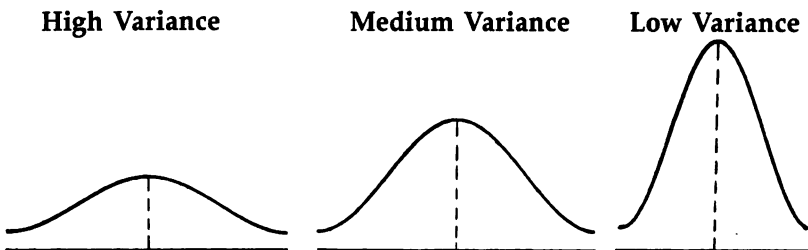
Figure 2. Normal Curve



The *mean* is the average value of the distribution, or the point where the curve would balance perfectly if held. The *mode* is the value of the distribution that occurs most often. And the *median* is the middle value: Half of all values lie to the left of the median, half to the right.

Actually, the normal curve isn't alone. There are countless variations, with each defined by just two parameters: *mean* and *variance*. Variance is a measure of the spread of the random numbers. A high variance scatters the numbers along a wider range, while a low variance "tightens" the spread. Take a look at Figure 3 for a moment. Each curve is normally distributed with the same mean but a *different* variance.

Figure 3. Variance and Curves



Number Generation

To use "Random Number Generator," type it in, save, and then run it. Enter the number of values you want to generate, 10, for example, followed by the mean and variance of the distribution. The computer does the rest. You'll see the ten numbers on the screen, printed out to three decimal places. Negative results are possible.

If you entered 10, then 5 for the mean, and 10 for the variance, you might see a list something like this:

```
8.639
8.098
4.994
5.176
6.132
2.160
-0.657
8.777
1.743
2.785
```

You can see that variance is not simply the range of the values, but does relate to the range. Keeping the same mean, but decreasing the variance to 2 would give you a different list. The values would all be closer to the mean.

If you've requested more than ten numbers to be generated, press any key to see the next group of values.

Random Number Generator

```
100 REM RANDOM NUMBER GENERATOR
110 REM INITIALIZE
120 GOSUB 200
130 REM ENTER MEAN & VARIANCE
140 GOSUB 390
150 REM GENERATE VALUES
160 GOSUB 500
170 REM DISPLAY VALUES
180 GOSUB 680
190 END
200 REM INITIALIZE
210 PRINT CHR$(21): TEXT : HOME
220 REM MAXIMUM NUMBER OF VALUES
230 DATA 300
240 READ NX
250 DIM X(NX)
260 REM GREETING
270 PRINT "This program generates random n
      umbers
```

```

280 PRINT "from a normal distribution.
290 VTAB 4: HTAB 1: PRINT "How many would
    you like to
300 VTAB 5: HTAB 10: PRINT CHR$ (7);: PRINT
    SPC( 15)
310 VTAB 5: HTAB 1: INPUT "produce ?";N$
320 N = INT ( VAL (N$))
330 IF N < = 0 THEN 300
340 IF N > NX THEN VTAB 20: HTAB 9: PRINT
    "Only ";NX;" are allowed !": GOTO 300
350 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT I
360 REM DIGITS AFTER DECIMAL
370 DA = 3
380 RETURN
390 REM MEAN & VARIANCE
400 HOME
410 PRINT "Please enter the mean and varia
    nce of
420 PRINT "your distribution."; CHR$ (7)
430 VTAB 4: HTAB 5: INPUT "Mean = ? ";M$
440 MEAN = VAL (M$)
450 VTAB 6: HTAB 14: PRINT SPC( 15): PRINT
    CHR$ (7)
460 VTAB 6: HTAB 1: INPUT "Variance = ? ";
    V$
470 VARIANCE = VAL (V$)
480 IF VARIANCE < 0 THEN VTAB 22: HTAB 4:
    PRINT "I need a non-negative variance
    !": GOTO 450
490 RETURN
500 REM GENERATE VALUES
510 HOME
520 VTAB 12: HTAB 15: FLASH : PRINT "GENER
    ATING";
530 NORMAL
540 FOR I = 1 TO N STEP 2
550 REM SELECT TWO NUMBERS FROM A UNIFORM
    DISTRIBUTION
560 A = RND (1)
570 B = RND (1)
580 REM MAKE INTERMEDIATE CALCULATIONS
590 C = 2 * A - 1
600 D = 2 * B - 1
610 SUM = C * C + D * D
620 IF SUM > = 1 THEN 560
630 REM DESIRED NUMBERS
640 X(I) = SQR (VARIANCE) * C * SQR ( - 2
    * LOG (SUM) / SUM) + MEAN
650 X(I + 1) = SQR (VARIANCE) * D * SQR (
    - 2 * LOG (SUM) / SUM) + MEAN

```

```

660 NEXT I
670 RETURN
680 REM DISPLAY VALUES
690 FOR L = 0 TO INT ((N - 1) / 10)
700 REM HEADING
710 GOSUB 760
720 REM BODY
730 GOSUB 850
740 NEXT L
750 RETURN
760 REM HEADING
770 HOME
780 PRINT L$
790 PRINT TAB( 10)"RANDOM NORMAL NUMBERS
800 VTAB 4: HTAB 17: PRINT "Mean = ";MEAN
810 VTAB 5: HTAB 13: PRINT "Variance = ";V
  ARIANCE
820 PRINT L$
830 PRINT
840 RETURN
850 REM 10 OBSERVATIONS AT A TIME
860 FOR J = 1 TO 10
870 Q = J + L * 10
880 IF Q < = N THEN PRINT "No.( ";Q; TAB(
  10)"")="";:HT = 27:NR = X(Q): GOSUB 9000

890 NEXT J
900 VTAB 22: HTAB 1: PRINT L$
910 PRINT TAB( 14)"Press any key ";
920 GET S$
930 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
  00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
  1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
  L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
  Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
  "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
  ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

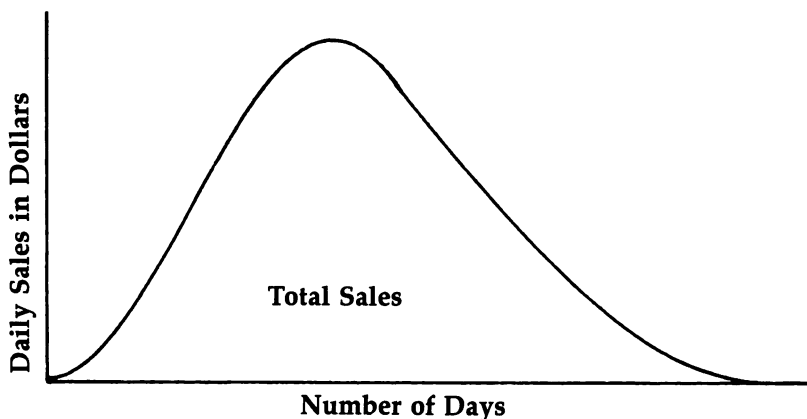
Numerical Integration

Creating graphs, whether to represent calories consumed per day, rainfall per month, or production per hour, can quickly illustrate a trend. But you may want to calculate the total value of that variable. To determine the value, all you have to do is compute the area under the curve. That's simple with this program.

When you draw a graph to illustrate a comparison, you normally end up with some sort of curve. For instance, if you compare sales to days of the month, you'll graph a curve. If the curve rises, you know that sales have risen, too. But often you'll want to know the *total* sales, not just the day-to-day variations. That figure is easy to find; it's the *area* under the curve. The process of computing this area is called *integration*, and the area is called the *integral*.

Take a look at Figure 1. It's an illustration of our sales example. Daily sales were plotted against the number of days.

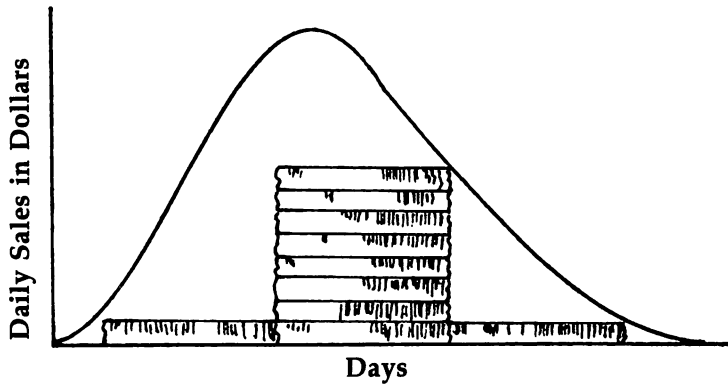
Figure 1. Sales



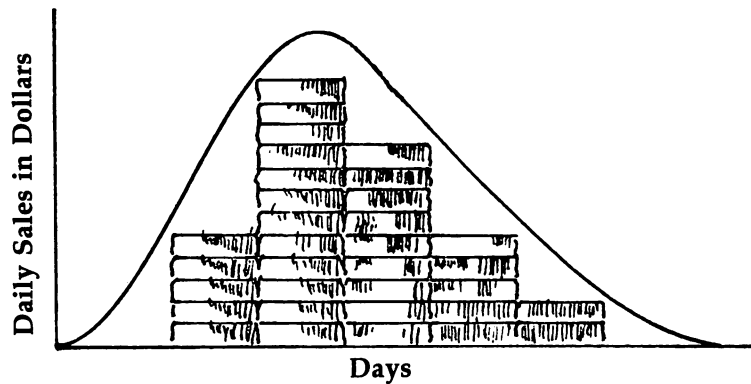
The region under the curve is the total sales. You could estimate this value by stuffing the curve with silver dollars, as the top drawing in Figure 2 shows. The number of coins represents the volume of sales.

Figure 2. Silver Dollars and Nickels

Silver Dollars



Nickels

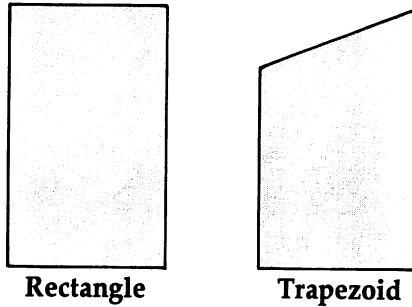


The approximation is pretty crude; there's a lot of unfilled space under the curve. You could reduce this error by using a smaller coin, perhaps a nickel instead of a dollar. That's illustrated by the bottom section of Figure 2. The smaller the coin, the better your estimate, because more area under the curve is filled. A coin of infinitely small size would yield a perfect estimate.

This coin-stuffing process is similar to what your computer does in *integrating a function*, or computing the area under a curve. Instead of using coins, however, the computer

uses lots and lots of trapezoids. A trapezoid is a four-sided figure (sometimes called a quadrilateral), with two parallel sides. Figure 3 shows you such a shape.

Figure 3. Trapezoid



By using trapezoids, the computer can better estimate the area. A rectangle, which could have been used, still leaves a lot of unfilled space under a curve.

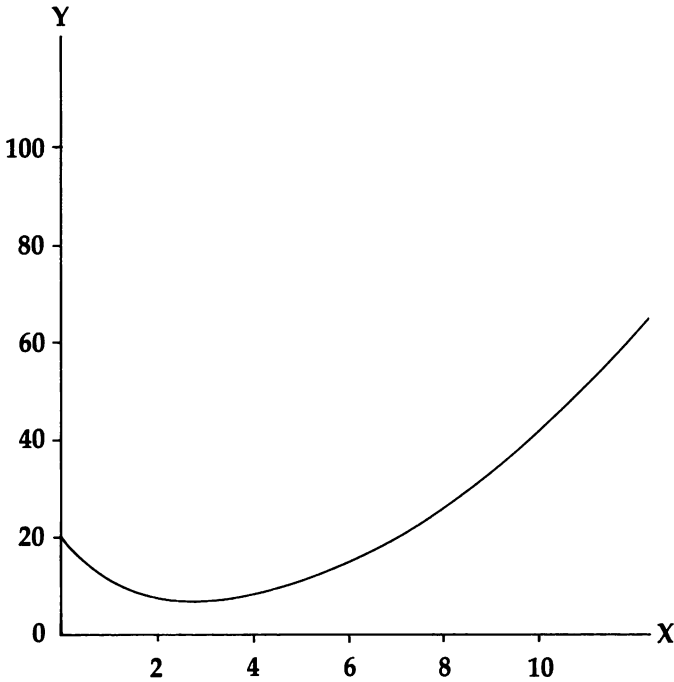
That's the basic geometry of the program. Here is how you can use it.

Integrate

After typing in and saving "Numerical Integration," run it. The program isn't hard to use, although you do have to have some information prepared before you sit down at the keyboard. The simplest explanation is an example.

Before you run the program, you need the formula for your curve. Perhaps it's simple, like $Y = X^2$. In our example, we'll use one a bit more complicated. How about $Y = 20 - 3X + X^2$. That's the normal notation. Unfortunately, you'll have to change your formula to something BASIC understands. Our example has to be changed to $Y = 20 - 3*X + X*X$. This formula would create a curve like the one shown in Figure 4.

Figure 4. Example Curve

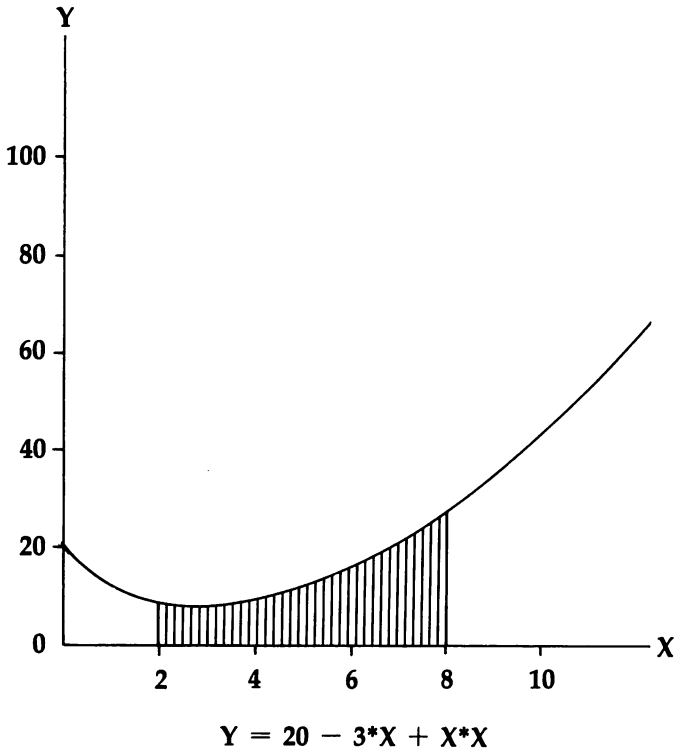


$$Y = 20 - 3X + X^2$$

Now you're ready to run Numerical Integration. The first screen asks if your curve's formula has been placed in line 190 of the program. In the example, the formula is already there, so you can press the Y key. If you were using your own formula, however, press the N key, type LIST 190, and enter your formula. Once it's in line 190, hit the RETURN key and rerun the program.

Next, you'll be asked to enter the lower and upper limits of the integration. This sets the left and right boundaries of the area to be calculated. Look at Figure 5 for a moment.

Figure 5. Example Limits



This is only a portion of the curve specified by the formula $Y = 20 - 3 \cdot X + X \cdot X$. If you wanted to compute the shaded area, you'd enter 2 for the lower (left) limit and 8 for the upper (right) limit. Type in those numbers at the appropriate prompts.

The program then asks for the number of intervals (the number of invisible trapezoids) to be used to figure the area bounded by the selected limits, the curve, and the x-axis (the bottom line of the graph). The more intervals, the more accurate the integration will be. There's a trade-off, unfortunately. The more intervals, the longer it takes the computer to calculate the area. Enter 3, and almost instantly the program returns the result. The integral is 202.00. That's just an approximation, though. To get a more accurate result, rerun the program, specifying the same limits, but this time using

200 trapezoids for measurement. The integral now shows as 198.001. That's only 0.001 off the mark.

The total of 198.001 might represent the sales from day 2 to day 8, a typical week. Not only can you use graphs as an illustration of something like sales, but with this program, you can compute totals as well.

Numerical Integration

```

100 REM NUMERICAL INTEGRATION
110 REM INITIALIZE
120 GOSUB 210
130 REM COMPUTE
140 GOSUB 720
150 REM DISPLAY
160 GOSUB 900
170 END
180 REM FUNCTION
190 Y = 20 - 3 * X + X * X
200 RETURN
210 REM INITIALIZE
220 REM TITLE
230 GOSUB 290
240 REM GREETING
250 GOSUB 380
260 REM LIMITS OF INTEGRATION
270 GOSUB 510
280 RETURN
290 REM TITLE
300 PRINT CHR$(21): TEXT : HOME
310 VTAB 12: HTAB 15: PRINT "Numerical
320 VTAB 13: HTAB 16: PRINT "Integration
330 FOR TIME = 1 TO 1000: NEXT TIME
340 L$ = "": FOR I = 1 TO 39: L$ = L$ + "=":
    NEXT I
350 REM DIGITS AFTER DECIMAL
360 DA = 3
370 RETURN
380 REM GREETING
390 HOME
400 PRINT "This program tallies the defini
    te
410 PRINT "integral of a function using th
    e
420 PRINT "trapezoidal rule.
430 PRINT
440 PRINT "Is your function now in line 19
    0 of
450 PRINT "the program (Y/N) ?";
    
```

```

460 PRINT CHR$ (7);
470 GET S$
480 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 460
490 IF S$ = "N" OR S$ = "n" THEN VTAB 12:
    HTAB 1: PRINT "Please put it there, a
    nd RUN again.": STOP
500 RETURN
510 REM LIMITS
520 HOME
530 PRINT "Please enter the limits of inte
    gration:
540 VTAB 3: HTAB 18: PRINT SPC( 15)
550 VTAB 5: HTAB 18: PRINT SPC( 15)
560 VTAB 3: HTAB 10: PRINT "Lower = "; CHR$
    (7);: INPUT LW$
570 VTAB 5: HTAB 10: PRINT "Upper = "; CHR$
    (7);: INPUT UP$
580 LOWER = VAL (LW$)
590 UPPER = VAL (UP$)
600 IF UPPER < = LOWER THEN VTAB 20: HTAB
    5: PRINT "Make Upper greater than Lowe
    r !": GOTO 540
610 REM NUMBER OF INTERVALS TO USE
620 GOSUB 640
630 RETURN
640 REM NUMBER OF INTERVALS
650 VTAB 20: HTAB 1: PRINT SPC( 39)
660 VTAB 10: HTAB 1: PRINT "How many inter
    vals within these limits
670 VTAB 11: HTAB 22: PRINT SPC( 15); CHR$
    (7)
680 VTAB 11: HTAB 1: INPUT "do you want to
    use ? ";N$
690 N = INT ( VAL (N$))
700 IF N < = 0 THEN VTAB 20: HTAB 8: PRINT
    "Come on ! That's too few.": GOTO 670
710 RETURN
720 REM COMPUTE
730 HOME
740 FLASH : VTAB 12: HTAB 15: PRINT "INTEG
    RATING";: NORMAL
750 REM INTERVAL SIZE
760 DT = (UPPER - LOWER) / N
770 REM BASIC CALCULATION
780 SUM = 0: X = LOWER
790 FOR I = 0 TO N
800 GOSUB 190
810 SUM = SUM + Y

```

```

820 X = X + DT
830 NEXT I
840 REM COMPUTE END VALUES AGAIN
850 X = LOWER: GOSUB 190:E1 = Y
860 X = UPPER: GOSUB 190:E2 = Y
870 REM AREA
880 AREA = (SUM - .5 * (E1 + E2)) * DT
890 RETURN
900 REM DISPLAY RESULT
910 HOME
920 PRINT L$
930 PRINT TAB( 9)"RESULTS OF INTEGRATION
940 PRINT L$
950 VTAB 5: HTAB 11: PRINT "Lower Limit ="
    ;:HT = 38:NR = LOWER: GOSUB 9000
960 VTAB 7: HTAB 11: PRINT "Upper Limit ="
    ;:NR = UPPER: GOSUB 9000
970 VTAB 9: HTAB 3: PRINT "Number of Inter
    vals =";:NR = N: GOSUB 9000
980 VTAB 10: HTAB 1: PRINT "Size of Each I
    nterval =";:NR = DT: GOSUB 9000
990 VTAB 12: HTAB 14: PRINT "Integral =";:
    NR = AREA: GOSUB 9000
1000 VTAB 22: HTAB 1: PRINT L$
1010 PRINT TAB( 14)"Press any key ";
1020 GET S$
1030 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
    00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
    1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
    L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
    Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Derivative

Ironically, it seems the only constant in the world is change. People age, matter decays, and the stock market fluctuates. This program shows how you can measure that rate of change.

A *derivative* measures the rate of change of a function. A *function*, in turn, is simply a mapping of two variables. An example of two such variables might be ice cream sales and months of the year. Sales depend upon the season. The number of cones sold increases once winter ends, peaks during the summer, and then falls off when autumn begins. This is the same as saying that the derivative is positive from January through July (as sales rise) and negative from August through December (as sales fall).

With your computer and "Derivative," you can calculate this rate of change. The program enables you to measure the rate of change of a variable at any point in time.

Sales Versus Season

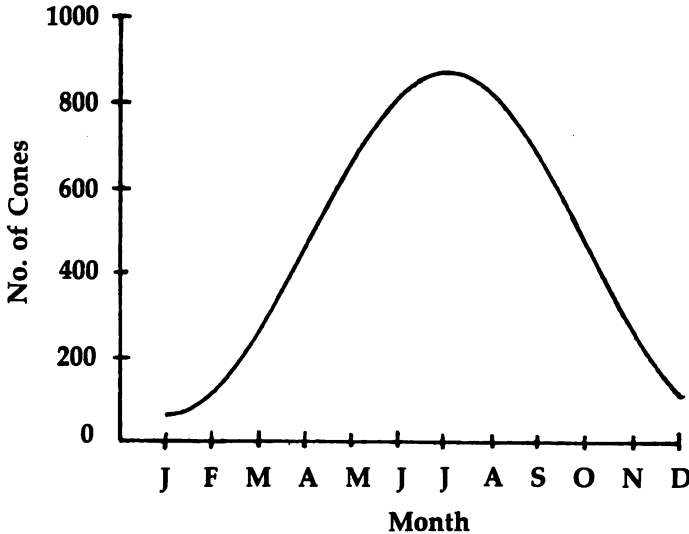
Run the program after you've typed it in and saved it. As in the previous program, "Numerical Integration," you need to have the formula of the curve before you can do the computations. Once you have that, it's a simple process to calculate the derivative.

An example might help. Glance at the figure; it's a graph showing a rough representation of the function of ice cream cones sold as related to months of the year. Note how sales increase as summer approaches.

Unlike when we used Numerical Integration, we don't want to know how many cones we sold during the entire year—instead, we want to know the rate of change in sales for the month of June. It's pretty easy to compute.

Make sure you have the formula for the curve in syntax BASIC can understand. For instance, X^2 should be printed as $X*X$. When you run the program, it will ask if the curve's formula has been inserted in line 200. If it hasn't, press the N key, type LIST 200, and enter the formula. Press RETURN and rerun the program. If you've already put the formula into the program, press the Y key and the screen display will change.

Ice Cream Cones



Let's change what shows in the listing to a formula representing the curve you saw in the figure. Modify line 200 so that it reads:

$$Y = -250 + 330 * X - 25 * X * X$$

(A more standard notation for this would be $Y = -250 + 330X - 25X^2$.) The Apple evaluates expressions more accurately when they are expressed as multiplied values than it does when they are expressed using exponents. Thus, $X * X$ is better than $X \uparrow X$.

Next, tell the program where on the curve to evaluate the derivative. In our example, June is the sixth month. Enter 6. The computer responds with a value of 29.998. Since it's positive, it means that sales are increasing at the rate of roughly 30 cones per month in June. Try entering other values to evaluate other parts of the curve.

As long as you know the formula of the curve you want to evaluate, this program makes finding derivatives easy and fast.

Derivative

```
100 REM DERIVATIVE
110 REM INITIALIZE
120 GOSUB 220
```

```

130 REM COMPUTE
140 GOSUB 520
150 REM DISPLAY
160 GOSUB 730
170 IF AN$ = "Y" OR AN$ = "y" THEN 140
180 END
190 REM FUNCTION
200 Y = 3 + 5 * X * X
210 RETURN
220 REM INITIALIZE
230 REM TITLE
240 GOSUB 280
250 REM GREETING
260 GOSUB 360
270 RETURN
280 REM TITLE
290 PRINT CHR$ (21): TEXT : HOME
300 VTAB 12: HTAB 15: PRINT "Derivative ";
310 FOR TIME = 1 TO 1000: NEXT
320 L$ = "": FOR I = 1 TO 39: L$ = L$ + "=":
    NEXT I
330 REM DIGITS TO RIGHT OF DECIMAL
340 DA = 3
350 RETURN
360 REM GREETING
370 HOME
380 PRINT "This program evaluates the derivative
390 PRINT "of the function  $Y = f(X)$ , at an
    y value
400 PRINT "of X.
410 PRINT
420 PRINT "It does this by computing the ratio
430 PRINT "Delta Y/Delta X for a tiny value of
440 PRINT "Delta X.
450 PRINT CHR$ (7)
460 PRINT "Is your function now in line
470 PRINT "number 200 (Y/N) ? ";
480 GET S$
490 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 480
500 IF S$ = "N" OR S$ = "n" THEN VTAB 15:
    HTAB 1: PRINT "Please put it there, and RUN again.": STOP
510 RETURN
520 REM COMPUTE
530 REM ENTER VALUE

```

```

540 GOSUB 580
550 REM COMPUTE
560 GOSUB 660
570 RETURN
580 REM ENTER VALUE
590 HOME
600 PRINT "Please enter the value at which
    you
610 PRINT "want the derivative evaluated.
620 PRINT CHR$ (7)
630 INPUT "X = ? ";X$
640 X0 = VAL (X$)
650 RETURN
660 REM COMPUTE
670 X = X0: GOSUB 200:Y0 = Y
680 REM VALUE OF FUNCTION AT X1
690 X = X0 + 1E - 04: GOSUB 200:X1 = X:Y1 =
    Y
700 REM DERIVATIVE
710 DERV = (Y1 - Y0) / (X1 - X0)
720 RETURN
730 REM DISPLAY
740 HOME
750 PRINT L$
760 PRINT TAB( 7)"RESULTS OF DIFFERENTIAT
    ION
770 PRINT L$
780 VTAB 6: HTAB 1: PRINT "For X"; TAB( 12
    )"="";HT = 27:NR = X0: GOSUB 9000
790 VTAB 8: HTAB 1: PRINT "Derivative =";
    NR = DERV: GOSUB 9000
800 VTAB 20: HTAB 1: PRINT L$
810 VTAB 22: HTAB 9: PRINT "Evaluate again
    (Y/N) ? "; CHR$ (7);
820 GET AN$
830 IF AN$ < > "Y" AND AN$ < > "y" AND A
    N$ < > "N" AND AN$ < > "n" THEN 820
840 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
    00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
    1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
    L) * 10 ^ DA)

```

```
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
      Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
      "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
      ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN
```


Matrix Addition and Subtraction

Mathematics is the Queen of Science, and
Arithmetic the Queen of Mathematics.
—Karl Friedrich Gauss

This program, along with the next three, allows you to manipulate *matrices*, which are groups of numbers or variables arranged in rectangular fashion and enclosed in brackets. They provide a shorthand method for calculating several variables or numbers at once and are often used to solve simultaneous equations. The members of a matrix are called its *elements*. As Figure 1 shows, these can be single numbers, variables, or expressions (combinations of numbers and variables).

Figure 1. Matrices

$$\begin{array}{ccc} \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} & \begin{bmatrix} a & b \\ c & d \end{bmatrix} & \begin{bmatrix} (3 + a) & (4 * c) \\ (x + y) & (b / d) \end{bmatrix} \\ A & B & C \end{array}$$

The elements of matrices A, B, and C are numbers, variables, and expressions, respectively.

The size of a matrix is denoted by its dimensions. A matrix with only one dimension is called a *row* or *column vector*. A matrix of two dimensions, on the other hand, generally contains several rows and several columns. And a matrix of three dimensions, as Figure 2 shows, contains rows, columns, and layers, corresponding to height, width, and depth.

In specifying the size of a two-dimensional array, the expression $R \times C$ is used, where R is the number of rows and C the number of columns. Hence, a matrix with 3 rows and 7 columns is of the order 3×7 . A matrix with 10 rows and 8 columns is of the order 10×8 .

Figure 2. Different Dimensions

One dimension

$$\begin{bmatrix} 6 & 7 & 2 \end{bmatrix}$$

1 row \times 3 columns

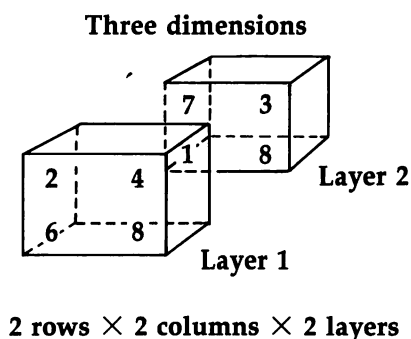
$$\begin{bmatrix} 6 \\ 7 \\ 2 \end{bmatrix}$$

3 rows \times 1 column

Two dimensions

$$\begin{bmatrix} 5 & 3 & 7 \\ 8 & 9 & 1 \end{bmatrix}$$

2 rows \times 3 columns



Adding and Subtracting

You can add or subtract matrices only when they're the same size. The sum of two matrices is the sum of the corresponding elements of each. And the difference is the difference between corresponding elements. Look at Figure 3, which illustrates how the arithmetic works.

Figure 3. Matrix Addition and Subtraction

Addition:

$$\begin{array}{ccc}
 \begin{bmatrix} 5 & 3 \\ 7 & 0 \end{bmatrix} & + & \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} (5 + 2) & (3 + 4) \\ (7 + 1) & (0 + 3) \end{bmatrix} = \begin{bmatrix} 7 & 7 \\ 8 & 3 \end{bmatrix} \\
 \text{A} & & \text{B} \qquad \qquad \qquad \text{C}
 \end{array}$$

Subtraction:

$$\begin{array}{ccc}
 \begin{bmatrix} 5 & 3 \\ 7 & 0 \end{bmatrix} & - & \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} (5 - 2) & (3 - 4) \\ (7 - 1) & (0 - 3) \end{bmatrix} = \begin{bmatrix} 3 & -1 \\ 6 & -3 \end{bmatrix} \\
 \text{A} & & \text{B} \qquad \qquad \qquad \text{C}
 \end{array}$$

As a general rule, $C(I,J) = A(I,J) + B(I,J)$, where I represents row I of a matrix and J represents column J .

When adding matrices, the value in the first row and first column of matrix A(5) is added to the corresponding element in matrix B(2). Thus, $5 + 2 = 7$, the result of which shows in the first row, first column of matrix C. Subtraction is handled in the same way: $5 - 2 = 3$, the value you see in the first row, first column of matrix C, in the bottom part of Figure 3.

To solve such two-dimensional matrix addition and/or subtraction, all you need to do is type in, save, and run this program. Matrices up to 15×25 are allowed. If you want to add or subtract larger matrices, you need to alter line 310. Allowing 25 rows and 25 columns would necessitate changing line 310 to read:

310 DATA 25,25

Enter the number of rows and columns for your matrices, then key in the elements of each matrix. The program even allows you to review and edit your entries. Once all the values have been entered, you can either add or subtract. In the latter, the second matrix is always subtracted from the first.

This program is especially helpful when you're trying to add or subtract large matrices. It's fast and it's accurate.

Matrix Addition and Subtraction

```

100 REM MATRIX ADDITION & SUBTRACTION
110 REM INITIALIZE
120 GOSUB 220
130 REM ENTER DATA
140 GOSUB 690
150 REM EDIT DATA
160 GOSUB 820
170 REM COMPUTE
180 GOSUB 1170
190 REM DISPLAY RESULT
200 GOSUB 1490
210 END
220 REM INITIALIZE
230 REM HEADING
240 GOSUB 280
250 REM ENTER ORDER
260 GOSUB 470
270 RETURN
280 REM HEADING
290 PRINT CHR$(21): TEXT : HOME
300 REM MAX SIZE
310 DATA 15,25
320 READ RX,CX
330 DIM X(2,RX,CX)
340 PRINT "This program adds or subtracts
any two
350 PRINT "matrices. The maximum allowabl
e size
360 PRINT "of each is
370 VTAB 5: HTAB 14: PRINT "Rows = ";RX
380 VTAB 6: HTAB 11: PRINT "Columns = ";CX

390 PRINT
400 PRINT "Change line 310 for different v
alues.
410 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
NEXT I
420 REM DIGITS TO RIGHT OF DECIMAL
430 DA = 3
440 VTAB 23: HTAB 14: PRINT "Press any key
";
450 GET S$
460 RETURN
470 REM ORDER
480 REM ROWS
490 GOSUB 530
500 REM COLUMNS
510 GOSUB 610

```

```

520 RETURN
530 REM ROWS
540 HOME
550 VTAB 1: HTAB 28: PRINT SPC( 10); CHR$
    (7)
560 VTAB 1: HTAB 1: INPUT "How many rows d
    o you have ?";R$
570 R = INT ( VAL (R$))
580 IF R < 1 THEN 550
590 IF R > RX THEN VTAB 20: HTAB 7: PRINT
    "Sorry, only ";RX;" are allowed!": GOTO
    550
600 RETURN
610 REM COLUMNS
620 VTAB 20: HTAB 7: PRINT SPC( 30)
630 VTAB 3: HTAB 19: PRINT SPC( 10); CHR$
    (7)
640 VTAB 3: HTAB 1: INPUT "How many column
    s ?";C$
650 C = INT ( VAL (C$))
660 IF C < 1 THEN 630
670 IF C > CX THEN VTAB 20: HTAB 7: PRINT
    "Sorry, only ";CX;" are allowed!": GOTO
    630
680 RETURN
690 REM ENTER DATA
700 T$(1) = "FIRST":T$(2) = "SECOND"
710 FOR Q = 1 TO 2
720 HOME : PRINT CHR$ (7);
730 PRINT "Please enter data on the ";T$(Q
    );" matrix.
740 FOR I = 1 TO C
750 VTAB 4: HTAB 1: PRINT "Column No.";I;"
    : "
760 FOR J = 1 TO R
770 VTAB 6: HTAB 4: PRINT "Row No.";J;" =
    ";:COL = POS (K) + 1
780 PRINT SPC( 15);: VTAB 6: HTAB COL: INPUT
    V$
790 X(Q,J,I) = VAL (V$)
800 NEXT J,I,Q
810 RETURN
820 REM EDIT DATA
830 FOR Z = 1 TO 2
840 FOR I = 1 TO C
850 FOR L = 0 TO INT ((R - 1) / 10)
860 REM DISPLAY DATA
870 GOSUB 920
880 REM CORRECT DATA
890 GOSUB 1030

```

```

900 NEXT L,I,Z
910 RETURN
920 REM DISPLAY
930 HOME
940 PRINT "These are values of the ";T$(Z)
    ;" matrix.
950 PRINT
960 PRINT TAB( 3)"Column ";I;":"
970 PRINT
980 FOR J = 1 TO 10
990 M = J + L * 10
1000 IF M < = R THEN PRINT "Row No. ";M; TAB(
    11)"=" ;X(Z,M,I)
1010 NEXT J
1020 RETURN
1030 REM CORRECT
1040 VTAB 16: HTAB 1: PRINT "Corrections (
    Y/N) ?"; CHR$(7);
1050 GET S$
1060 IF S$ = "N" OR S$ = "n" THEN 1160
1070 IF S$ < > "Y" AND S$ < > "y" THEN 1
    040
1080 VTAB 18: HTAB 19: PRINT SPC( 15); CHR$(
    7)
1090 VTAB 18: HTAB 1: INPUT "Row to correc
    t = ? ";Q$
1100 Q = INT ( VAL (Q$))
1110 IF Q < (1 + L * 10) OR Q > R OR Q > (
    10 + L * 10) THEN VTAB 20: HTAB 5: PRINT
    "Row outside of bounds shown !": GOTO
    1080
1120 VTAB 20: HTAB 5: PRINT SPC( 35)
1130 VTAB 20: HTAB 1: INPUT "What should t
    he value be ? ";S$
1140 X(Z,Q,I) = VAL (S$)
1150 GOSUB 920: GOTO 1040
1160 RETURN
1170 REM COMPUTE
1180 REM ENTER CHOICE
1190 GOSUB 1230
1200 REM COMPUTE
1210 ON CHOICE GOSUB 1370,1430
1220 RETURN
1230 REM ENTER CHOICE
1240 HOME
1250 PRINT "Would you like to
1260 INVERSE
1270 VTAB 3: HTAB 10: PRINT "1"
1280 VTAB 5: HTAB 10: PRINT "2"
1290 NORMAL

```

```

1300 VTAB 3: HTAB 12: PRINT "Add or
1310 VTAB 5: HTAB 12: PRINT "Subtract
1320 VTAB 7: HTAB 1: PRINT "your matrices
    ? "; CHR$ (7);
1330 GET S$
1340 CHOICE = VAL (S$)
1350 IF CHOICE < > 1 AND CHOICE < > 2 THEN
    1320
1360 RETURN
1370 REM ADD
1380 FOR I = 1 TO R
1390 FOR J = 1 TO C
1400 X(1,I,J) = X(1,I,J) + X(2,I,J)
1410 NEXT J,I
1420 RETURN
1430 REM SUBTRACT
1440 FOR I = 1 TO R
1450 FOR J = 1 TO C
1460 X(1,I,J) = X(1,I,J) - X(2,I,J)
1470 NEXT J,I
1480 RETURN
1490 REM DISPLAY RESULTS
1500 FOR Q = 1 TO R STEP 10
1510 FOR I = 1 TO C STEP 2
1520 REM HEADING
1530 GOSUB 1580
1540 REM BODY
1550 GOSUB 1810
1560 NEXT I,Q
1570 RETURN
1580 REM HEADING
1590 HOME
1600 PRINT L$
1610 IF CHOICE = 1 THEN VTAB 2: HTAB 15: PRINT
    "MATRIX SUM
1620 IF CHOICE = 2 THEN VTAB 2: HTAB 12: PRINT
    "MATRIX DIFFERENCE
1630 PRINT L$
1640 REM COLUMN HEADING
1650 COL = 14
1660 FOR L = I TO I + 1
1670 IF L < = C THEN VTAB 5: HTAB COL: PRINT
    "Column ";L
1680 COL = COL + 16
1690 NEXT L
1700 REM BRACKETS
1710 INVERSE
1720 FOR M = 7 TO 18 STEP 11
1730 VTAB M: HTAB 7: PRINT CHR$ (32); HTAB
    (38): PRINT CHR$ (32)

```

```

1740 NEXT M
1750 FOR M = 7 TO 18
1760 VTAB M: HTAB 6: PRINT CHR$ (32)
1770 VTAB M: HTAB 39: PRINT CHR$ (32)
1780 NEXT M
1790 NORMAL
1800 RETURN
1810 REM VALUES
1820 ROW = 8
1830 FOR J = Q TO Q + 9
1840 IF J > R THEN 1920
1850 VTAB ROW: HTAB 1: PRINT "R ";J;
1860 COL = 21
1870 FOR L = I TO I + 1
1880 IF L < = C THEN VTAB ROW:HT = COL:N
    R = X(1,J,L): GOSUB 9000
1890 COL = COL + 16
1900 NEXT L
1910 ROW = ROW + 1
1920 NEXT J
1930 VTAB 22: HTAB 1: PRINT L$
1940 VTAB 23: HTAB 14: PRINT "Press any ke
Y ";
1950 GET S$
1960 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
000000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```


Matrix Multiplication

Multiplying matrices by hand is a bit more complex than simply adding or subtracting them. With this program, however, it's almost as easy.

You'll find this program even more valuable than the previous one, because multiplying two-dimensional matrices by hand is far more involved than just adding or subtracting them. With this program, all you have to enter are the matrices' sizes and their elements—the computer does the rest.

Before you begin to use "Matrix Multiplication," however, you must first determine if a matrix product exists. In the expression $X * Y$, X is considered the *lead* matrix, and Y the *lag* matrix. Two matrices can be multiplied only if the number of columns in the lead matrix equals the number of rows in the lag matrix.

It's easy to tell if a product exists. Jot down the dimensions of each array, side by side. If X is of the order 3×4 (3 rows and 4 columns), for instance, and Y is of the order 4×7 , you'd write

$$\begin{array}{ccc} X & * & Y \\ (3 \times 4) & & (4 \times 7) \end{array}$$

and see that multiplication is possible because the two middle dimensions (both 4 in this case) are equal. If you turned the matrices around, however, and tried to multiply Y by X ($Y * X$), a product is not possible. That's because the seven columns of the lead matrix (Y) do not match the three rows of the lag matrix (X).

The product of X and Y is computed by adding the products of row elements of matrix X and corresponding column elements of matrix Y for all rows and columns. As Figure 1 shows, this is a lot easier than it sounds.

Take a closer look at the top part of Figure 1, in which two vectors are multiplied. The first element of matrix A is multiplied by the first element of matrix B ($5 * 7$) and added to the product of the second elements of both matrices ($3 * 3$). The total is thus $35 + 9$, or 44. Note, too, that the middle dimensions (the 2's) cancel out, leaving a 1×1 matrix.

Figure 1. Matrix Multiplication

Two Vectors

$$\begin{array}{c}
 \left[\begin{array}{cc} 5 & 3 \end{array} \right] \quad (1 \times 2) \\
 \text{A}
 \end{array}
 *
 \begin{array}{c}
 \left[\begin{array}{c} 7 \\ 3 \end{array} \right] \quad (2 \times 1) \\
 \text{B}
 \end{array}
 =
 \left[\begin{array}{c} 5*7 + 3*3 \\ \end{array} \right] \quad (1 \times 1)
 =
 \left[\begin{array}{c} 44 \end{array} \right]$$

Two Matrices

$$\begin{array}{c}
 \left[\begin{array}{ccc} 2 & 3 & 3 \\ 0 & 5 & 1 \end{array} \right] \quad (2 \times 3) \\
 \text{A}
 \end{array}
 *
 \begin{array}{c}
 \left[\begin{array}{cc} 2 & 7 \\ 1 & 8 \\ 9 & 4 \end{array} \right] \quad (3 \times 2) \\
 \text{B}
 \end{array}
 =
 \begin{array}{c}
 \left[\begin{array}{cc} 2*2 + 3*1 + 3*9 & 2*7 + 3*8 + 3*4 \\ 0*2 + 5*1 + 1*9 & 0*7 + 5*8 + 1*4 \end{array} \right] \quad (2 \times 2) \\
 \text{Product of row 1 and column 1}
 \end{array}
 =
 \left[\begin{array}{cc} 34 & 50 \\ 14 & 44 \end{array} \right]$$

That kind of multiplication is fairly obvious. But consider the second example in Figure 1, where a 2×3 matrix is multiplied by a 3×2 matrix. It's the same process as above, just a bit more complicated. The first row in matrix A is multiplied, element by element, by each of the two columns in matrix B. Then the second row of A is multiplied by both columns of B. The results are listed as shown. Again, the middle dimensions (the 3's) cancel out, leaving a 2×2 matrix.

When you're working with matrices identical in size, the matrix product of $X * Y$ will not equal the product of $Y * X$. Figure 2 demonstrates this.

Figure 2. Unequal Products

$$\begin{array}{cc} \begin{bmatrix} 2 & 3 \\ 0 & 4 \end{bmatrix} & * & \begin{bmatrix} 1 & 0 \\ 2 & 7 \end{bmatrix} = \begin{bmatrix} (2*1 + 3*2) & (2*0 + 3*7) \\ (0*1 + 4*2) & (0*0 + 4*7) \end{bmatrix} = \begin{bmatrix} 8 & 21 \\ 8 & 28 \end{bmatrix} \\ \text{X} & & \text{Y} \end{array}$$

but

$$\begin{array}{cc} \begin{bmatrix} 1 & 0 \\ 2 & 7 \end{bmatrix} & * & \begin{bmatrix} 2 & 3 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} (1*2 + 0*0) & (1*3 + 0*4) \\ (2*2 + 7*0) & (2*3 + 7*4) \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 34 \end{bmatrix} \\ \text{Y} & & \text{X} \end{array}$$

Multiplication at Last

This program is very similar to "Matrix Addition and Subtraction" in use and execution. Type it in, save it, and enter RUN. As with the previous program, there are default limits to the size of the matrices. To change these defaults, alter line 310 in the program accordingly. The second screen display asks for the sizes of the matrices. (Remember that they must be in the form $(A \times B)(B \times C)$ in order to create a product.)

Now all you have to do is enter the matrices' elements. Review and editing capabilities are provided. As soon as you've typed in the values, the program computes the matrix product $X * Y$. You have the totals in front of you—all without having to lift a pencil.

Matrix Multiplication

```

100 REM MATRIX MULTIPLICATION
110 REM INITIALIZE
120 GOSUB 220
130 REM ENTER DATA
140 GOSUB 870
150 REM EDIT DATA
160 GOSUB 1000
170 REM COMPUTE
180 GOSUB 1390
190 REM DISPLAY RESULT
200 GOSUB 1510
210 END
220 REM INITIALIZE
230 REM HEADING
240 GOSUB 280
    
```

```

250 REM ENTER MATRIX SIZES
260 GOSUB 480
270 RETURN
280 REM HEADING
290 PRINT CHR$(21): TEXT : HOME
300 REM MAX SIZE
310 DATA 15,15
320 READ MR,MC
330 DIM X(MR,MC),Y(MR,MC),P(MR,MC)
340 PRINT "This program multiplies two mat
rices,
350 PRINT "X and Y. That is, the matrix p
roduct
360 PRINT "X*Y is tallied. The maximum al
lowable
370 PRINT "size of each array is
380 VTAB 6: HTAB 14: PRINT "Rows = ";MR
390 VTAB 7: HTAB 11: PRINT "Columns = ";MC

400 PRINT
410 PRINT "Change line 310 for different v
alues.
420 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
NEXT I
430 REM DIGITS AFTER DECIMAL
440 DA = 3
450 VTAB 23: HTAB 14: PRINT "Press any key
";
460 GET S$
470 RETURN
480 REM ENTER MATRIX SIZES
490 M$(1) = "X":M$(2) = "Y"
500 FOR I = 1 TO 2
510 GOSUB 570
520 NEXT I
530 REM CHECK CONFORMABILITY
540 GOSUB 720
550 IF CN$ = "NONCONFORMABLE" THEN 500
560 RETURN
570 REM MATRIX SIZES
580 HOME
590 PRINT "Please enter the size of matrix
";M$(I);"."
600 VTAB 3: HTAB 21: PRINT SPC( 15); CHR$(
7)
610 VTAB 3: HTAB 14: INPUT "Rows = ";R$
620 R(I) = VAL (R$)
630 IF R(I) < 1 THEN 600

```

```

640 IF R(1) > MR THEN VTAB 20: HTAB 4: PRINT
    "Sorry, only ";MR;" rows are allowed!"
    : GOTO 600
650 VTAB 20: HTAB 4: PRINT SPC( 36)
660 VTAB 5: HTAB 21: PRINT SPC( 15); CHR$
    (7)
670 VTAB 5: HTAB 11: INPUT "Columns = ";C$
680 C(I) = VAL (C$)
690 IF C(I) < 1 THEN 660
700 IF C(I) > MC THEN VTAB 20: HTAB 2: PRINT
    "Sorry, only ";MC;" columns are allowe
    d!": GOTO 660
710 RETURN
720 REM CHECK CONFORMABILITY
730 CN$ = ""
740 HOME
750 IF C(1) = R(2) THEN 860
760 PRINT "Sorry, I can't multiply your ar
    rays.
770 PRINT
780 PRINT "The number of columns in matrix
    X does
790 PRINT "NOT equal the number of rows in
    Y.
800 PRINT
810 PRINT "That is, ";C(1);" <> ";R(2);".
    Please try again."
820 PRINT
830 VTAB 23: HTAB 14: PRINT "Press any key
    ";
840 GET S$
850 CN$ = "NONCONFORMABLE"
860 RETURN
870 REM ENTER DATA
880 FOR Q = 1 TO 2
890 HOME : PRINT CHR$ (7);
900 PRINT "Please enter data on matrix ";M
    $(Q);"."
910 FOR I = 1 TO C(Q)
920 VTAB 4: HTAB 1: PRINT "Column No.";I;"
    ";
930 FOR J = 1 TO R(Q)
940 VTAB 6: HTAB 4: PRINT "Row No.";J;" =
    ";:COL = POS (K) + 1
950 PRINT SPC( 15);: VTAB 6: HTAB COL: INPUT
    V$
960 IF Q = 1 THEN X(J,I) = VAL (V$)
970 IF Q = 2 THEN Y(J,I) = VAL (V$)
980 NEXT J,I,Q

```

```

990 RETURN
1000 REM EDIT DATA
1010 FOR Z = 1 TO 2
1020 FOR I = 1 TO C(Z)
1030 FOR L = 0 TO INT ((R(Z) - 1) / 10)
1040 REM DISPLAY DATA
1050 GOSUB 1100
1060 REM CORRECT DATA
1070 GOSUB 1240
1080 NEXT L,I,Z
1090 RETURN
1100 REM DISPLAY
1110 HOME
1120 PRINT "These are values of matrix ";M
    $(Z);". "
1130 PRINT
1140 PRINT TAB( 3)"Column ";I;":"
1150 PRINT
1160 FOR J = 1 TO 10
1170 M = J + L * 10
1180 IF M > R(Z) THEN 1220
1190 IF Z = 1 THEN V = X(M,I)
1200 IF Z = 2 THEN V = Y(M,I)
1210 PRINT "Row No.";M; TAB( 11)"= ";V
1220 NEXT J
1230 RETURN
1240 REM CORRECT
1250 VTAB 16: HTAB 1: PRINT "Corrections (
Y/N) ?"; CHR$( 7);
1260 GET S$
1270 IF S$ = "N" OR S$ = "n" THEN 1380
1280 IF S$ < > "Y" AND S$ < > "y" THEN 1
250
1290 VTAB 18: HTAB 19: PRINT SPC( 15); CHR$(
7)
1300 VTAB 18: HTAB 1: INPUT "Row to correc
t = ? ";Q$
1310 Q = INT ( VAL (Q$))
1320 IF Q < (1 + L * 10) OR Q > R(Z) OR Q >
(10 + L * 10) THEN VTAB 20: HTAB 5: PRIN
"Row outside of bounds shown !": GOTO
1290
1330 VTAB 20: HTAB 5: PRINT SPC( 35)
1340 VTAB 20: HTAB 1: INPUT "What should t
he value be ? ";S$
1350 IF Z = 1 THEN X(Q,I) = VAL (S$)
1360 IF Z = 2 THEN Y(Q,I) = VAL (S$)
1370 GOSUB 1100: GOTO 1250
1380 RETURN

```

```

1390 REM COMPUTE
1400 HOME : FLASH
1410 VTAB 12: HTAB 15: PRINT "COMPUTING";
1420 NORMAL
1430 FOR TIME = 1 TO 500: NEXT TIME
1440 FOR I = 1 TO R(1)
1450 FOR J = 1 TO C(2)
1460 P(I,J) = 0
1470 FOR L = 1 TO C(1)
1480 P(I,J) = P(I,J) + X(I,L) * Y(L,J)
1490 NEXT L,J,I
1500 RETURN
1510 REM DISPLAY RESULTS
1520 FOR Q = 1 TO R(1) STEP 10
1530 FOR I = 1 TO C(2) STEP 2
1540 REM HEADING
1550 GOSUB 1600
1560 REM BODY
1570 GOSUB 1820
1580 NEXT I,Q
1590 RETURN
1600 REM HEADING
1610 HOME
1620 PRINT L$
1630 PRINT TAB( 14)"MATRIX PRODUCT
1640 PRINT L$
1650 REM COLUMN HEADING
1660 COL = 14
1670 FOR L = 1 TO I + 1
1680 IF L < = C(2) THEN VTAB 5: HTAB COL
: PRINT "Column ";L
1690 COL = COL + 16
1700 NEXT L
1710 REM BRACKETS
1720 INVERSE
1730 FOR M = 7 TO 18 STEP 11
1740 VTAB M: HTAB 7: PRINT CHR$( 32);: HTAB
(38): PRINT CHR$( 32)
1750 NEXT M
1760 FOR M = 7 TO 18
1770 VTAB M: HTAB 6: PRINT CHR$( 32)
1780 VTAB M: HTAB 39: PRINT CHR$( 32)
1790 NEXT M
1800 NORMAL
1810 RETURN
1820 REM VALUES
1830 ROW = 8
1840 FOR J = Q TO Q + 9
1850 IF J > R(1) THEN 1930
1860 VTAB ROW: HTAB 1: PRINT "R ";J;

```

```

1870 COL = 21
1880 FOR L = I TO I + 1
1890 IF L < = C(2) THEN VTAB ROW:HT = CO
      L:NR = P(J,L): GOSUB 9000
1900 COL = COL + 16
1910 NEXT L
1920 ROW = ROW + 1
1930 NEXT J
1940 VTAB 22: HTAB 1: PRINT L$
1950 VTAB 23: HTAB 14: PRINT "Press any ke
      y ";
1960 GET S$
1970 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "": ZS$ = "": ZD$ = "": ZZ$ = "0000
      00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
      1)
9060 LET ZL = INT (ZN): ZR = INT ((ZN - Z
      L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
      Z$ + STR$ (ZR), DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3, 1) < >
      "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
      ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```


Matrix Inversion

Yet another way to manipulate matrices is with the process of inversion. The explanation may sound complicated, but the program's not.

In arithmetic, the reciprocal, or *inverse*, of a number is a value which when multiplied by the original figure results in 1. For example, the inverse of 2 is $1/2$ (since $2 * 1/2 = 1$). In more general terms, the inverse of X is $1/X$.

In matrix mathematics, an inverse is similarly defined. Specifically, the inverse of matrix X is another matrix Y such that $X * Y = I$, where I is an *identity matrix*. An identity matrix is an array with 1's along the principal diagonal and 0's everywhere else. (The *principal diagonal* is an imaginary line running from the top-left corner of the matrix to the bottom-right corner.) An identity matrix is always square (the same number of rows and columns). This means that only square matrices can be inverted.

As mentioned in the previous article, the matrix product $X * Y$ will not usually equal the matrix product $Y * X$, even when both expressions are defined. One exception to this rule is when Y equals the inverse of X . In this case, Y is usually written X^{-1} (read *X inverse*), with $X * X^{-1} = X^{-1} * X = I$. Figure 1 shows this exception.

Figure 1. Matrix Inversion

$$\underbrace{\begin{bmatrix} 2 & 4 \\ 4 & 6 \end{bmatrix}}_X * \underbrace{\begin{bmatrix} -\frac{3}{2} & 1 \\ 1 & -\frac{1}{2} \end{bmatrix}}_{\text{Inverse}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_I$$

and

$$\underbrace{\begin{bmatrix} -\frac{3}{2} & 1 \\ 1 & -\frac{1}{2} \end{bmatrix}}_{\text{Inverse}} * \underbrace{\begin{bmatrix} 2 & 4 \\ 4 & 6 \end{bmatrix}}_X = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_I$$

(If you want to verify this, use the program "Matrix Multiplication" to find the product of matrix X and its inverse. Instead of entering $-3/2$ as the first element of the inverse matrix, use -1.5 . Change $-1/2$ to -0.5 as well. The matrix product should be identical to I in Figure 1 above.)

The inverse of a matrix can be computed several different ways. This program uses a simple method. First, the program tacks an identity matrix onto your matrix. Then the left half of this augmented matrix is transformed into I (identity matrix), using elementary row operations. What used to be the identity matrix is now the inverse. If this sounds complicated, don't worry; Figure 2 takes you through the procedure.

Figure 2. Inversion Mechanics

The program performs the following steps to invert a matrix.

1. Tack an identity matrix (I) onto the array that you want to invert (X).

$$\left[\begin{array}{cc|cc} 2 & 4 & 1 & 0 \\ 1 & 4 & 0 & 1 \end{array} \right]$$

$\underbrace{\hspace{4em}}_X$
 $\underbrace{\hspace{4em}}_I$

2. Transform X into I using elementary row operations.

- a. Divide the first row by the pivot element, 2. This gives you:

$$\left[\begin{array}{cc|cc} 1 & 2 & 1/2 & 0 \\ 1 & 4 & 0 & 1 \end{array} \right]$$

- b. Subtract the first row from the second, to result in:

$$\left[\begin{array}{cc|cc} 1 & 2 & 1/2 & 0 \\ 0 & 2 & -1/2 & 1 \end{array} \right]$$

- c. Divide the second row above by the pivot element, 2. The result is:

$$\left[\begin{array}{cc|cc} 1 & 2 & 1/2 & 0 \\ 0 & 1 & -1/4 & 1/2 \end{array} \right]$$

- d. Finally, multiply the second row by 2, and subtract the product from the first row. The final *inverse* matrix is on the right.

$$\left[\begin{array}{cc|cc} 1 & 0 & 1 & -1 \\ 0 & 1 & -1/4 & 1/2 \end{array} \right]$$

$\underbrace{\hspace{4em}}_{\text{Inverse}}$

Inversion Therapy

Fortunately, you don't have to know any of this to use the program. Just as with the other matrix programs in this chapter, all you have to do is set the size of the matrix and input the matrix elements. The inverse matrix is created and displayed by the computer.

Run the program and enter the size of the matrix you want to invert. Since the matrix has to be square, you just need to type in one number. (If you want to change the default size limitation, alter the DATA statement in line 310.) Type in the matrix's elements according to the prompts, and the program does the rest.

Matrix Inversion

```

100 REM MATRIX INVERSION
110 REM INITIALIZE
120 GOSUB 220
130 REM ENTER DATA
140 GOSUB 560
150 REM EDIT DATA
160 GOSUB 670
170 REM INVERT
180 GOSUB 1010
190 REM DISPLAY RESULT
200 GOSUB 1250
210 END
220 REM INITIALIZE
230 REM HEADING
240 GOSUB 280
250 REM ENTER ORDER
260 GOSUB 470
270 RETURN
280 REM HEADING
290 PRINT CHR$(21): TEXT : HOME
300 REM MAX SIZE
310 DATA 20
320 READ MX
330 DIM X(MX,2 * MX)
340 PRINT "This program inverts a matrix.
    The
350 PRINT "maximum allowable size of the a
    rray,"
360 PRINT "which must be square, is
370 VTAB 5: HTAB 14: PRINT "Rows = ";MX
380 VTAB 6: HTAB 11: PRINT "Columns = ";MX
390 PRINT
400 PRINT "Change line 310 for different v
    alues.
```

```

410 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT I
420 REM DIGITS AFTER DECIMAL
430 DA = 3
440 VTAB 23: HTAB 14: PRINT "Press any key
    ";
450 GET S$
460 RETURN
470 REM ORDER
480 HOME
490 PRINT "How many rows (or columns) does
    your
500 VTAB 2: HTAB 15: PRINT SPC( 20); CHR$
    (7)
510 VTAB 2: HTAB 1: INPUT "matrix have ? "
    ;R$
520 K = INT ( VAL (R$))
530 IF K < 1 THEN 500
540 IF K > MX THEN VTAB 20: HTAB 7: PRINT
    "Sorry, only ";MX;" are allowed!": GOTO
    500
550 RETURN
560 REM ENTER DATA
570 HOME
580 PRINT "Please enter data on your matri
    x.
590 FOR I = 1 TO K
600 VTAB 4: HTAB 1: PRINT "Column No.";I;"
    ";
610 FOR J = 1 TO K
620 VTAB 6: HTAB 4: PRINT "Row No.";J;" =
    ";:COL = POS (Z) + 1
630 PRINT SPC( 15);: VTAB 6: HTAB COL: INPUT
    V$
640 X(J,I) = VAL (V$)
650 NEXT J,I
660 RETURN
670 REM EDIT DATA
680 FOR I = 1 TO K
690 FOR L = 0 TO INT ((K - 1) / 10)
700 REM DISPLAY DATA
710 GOSUB 760
720 REM CORRECT DATA
730 GOSUB 870
740 NEXT L,I
750 RETURN
760 REM DISPLAY
770 HOME
780 PRINT "These are values of your matrix
790 PRINT

```

```

800 PRINT TAB( 3)"Column ";I;":"
810 PRINT
820 FOR J = 1 TO 10
830 M = J + L * 10
840 IF M < = K THEN PRINT "Row No.";M; TAB(
    11)"=" ;X(M,I)
850 NEXT J
860 RETURN
870 REM CORRECT
880 VTAB 16: HTAB 1: PRINT "Corrections (Y
    /N) ?"; CHR$( 7);
890 GET S$
900 IF S$ = "N" OR S$ = "n" THEN 1000
910 IF S$ < > "Y" AND S$ < > "y" THEN 88
    0
920 VTAB 18: HTAB 19: PRINT SPC( 15); CHR$(
    7)
930 VTAB 18: HTAB 1: INPUT "Row to correct
    = ? ";Q$
940 Q = INT ( VAL (Q$))
950 IF Q < (1 + L * 10) OR Q > K OR Q > (1
    0 + L * 10) THEN VTAB 20: HTAB 5: PRINT
    "Row outside of bounds shown !": GOTO
    920
960 VTAB 20: HTAB 5: PRINT SPC( 35)
970 VTAB 20: HTAB 1: INPUT "What should th
    e value be ? ";S$
980 X(Q,I) = VAL (S$)
990 GOSUB 760: GOTO 880
1000 RETURN
1010 REM INVERT
1020 HOME
1030 FLASH : VTAB 12: HTAB 15: PRINT "INVE
    RTING";: NORMAL
1040 FOR TIME = 1 TO 500: NEXT TIME
1050 REM TACK ON IDENTITY MATRIX
1060 FOR I = 1 TO K
1070 FOR J = 1 TO K
1080 X(I,K + J) = 0
1090 IF J = I THEN X(I,K + J) = 1
1100 NEXT J,I
1110 REM COMPUTE
1120 FOR I = 1 TO K
1130 REM ADJUST KEY ROW
1140 PIVOT = X(I,I)
1150 FOR J = I TO 2 * K
1160 X(I,J) = X(I,J) / PIVOT
1170 NEXT J
1180 REM ADJUST REMAINING ROWS
1190 FOR J = 1 TO K

```

```

1200 X = X(J,I)
1210 FOR L = I TO 2 * K
1220 IF J < > I THEN X(J,L) = X(J,L) - X *
    X(I,L)
1230 NEXT L,J,I
1240 RETURN
1250 REM DISPLAY RESULTS
1260 FOR Q = 1 TO K STEP 10
1270 FOR I = 1 TO K STEP 2
1280 REM HEADING
1290 GOSUB 1340
1300 REM BODY
1310 GOSUB 1560
1320 NEXT I,Q
1330 RETURN
1340 REM HEADING
1350 HOME
1360 PRINT L$
1370 PRINT TAB( 14)"MATRIX INVERSE
1380 PRINT L$
1390 REM COLUMN HEADING
1400 COL = 14
1410 FOR L = I TO I + 1
1420 IF L < = K THEN VTAB 5: HTAB COL: PRINT
    "Column ";L
1430 COL = COL + 16
1440 NEXT L
1450 REM BRACKETS
1460 INVERSE
1470 FOR M = 7 TO 18 STEP 11
1480 VTAB M: HTAB 7: PRINT CHR$ (32);: HTAB
    (38): PRINT CHR$ (32)
1490 NEXT M
1500 FOR M = 7 TO 18
1510 VTAB M: HTAB 6: PRINT CHR$ (32)
1520 VTAB M: HTAB 39: PRINT CHR$ (32)
1530 NEXT M
1540 NORMAL
1550 RETURN
1560 REM VALUES
1570 ROW = 8
1580 FOR J = Q TO Q + 9
1590 IF J > K THEN 1670
1600 VTAB ROW: HTAB 1: PRINT "R ";J;
1610 COL = 21
1620 FOR L = I TO I + 1
1630 IF L < = K THEN VTAB ROW:HT = COL:N
    R = X(J,K + L): GOSUB 9000
1640 COL = COL + 16

```

```

1650 NEXT L
1660 ROW = ROW + 1
1670 NEXT J
1680 VTAB 22: HTAB 1: PRINT L$
1690 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
1700 GET S$
1710 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "": ZS$ = "": ZD$ = "": ZZ$ = "0000
000000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN): ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR), DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3, 1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

MatMan

You've seen how to add, subtract, multiply, and even invert matrices. But what if you want to do several operations at once or perform manipulations of even greater complexity? "MatMan" is your answer.

If you have ever wanted to add two matrices, multiply the sum by a third, and invert the product, "MatMan" can help. MatMan is a jack of all trades. It can add, subtract, transpose, and invert any two matrices, X and Y. The result is stored in Z. You can even perform further operations using X or Y or Z.

Take a look at Figure 1. Don't be intimidated by its complexity. MatMan can handle it.

Figure 1. Complex Operations

$$\left[\begin{array}{c} 3* \\ X \end{array} \right] \left\{ \left[\begin{array}{cc} 7 & 17 \\ -20 & 512 \end{array} \right] + \left[\begin{array}{cc} 32 & 9 \\ 1 & 18 \end{array} \right] \right\} * \left[\begin{array}{c} 2 \\ 2 \end{array} \right]$$

Y

Three distinct operations are required to evaluate this: addition, scalar multiplication (multiplying by a single number, rather than a matrix), and matrix multiplication. Want to try calculating this by hand? Let MatMan do it instead.

MatMan Menus

After typing in and saving MatMan, run it. The default limit for both the X and Y matrices is 10; it can be changed by altering line 490 in the program.

Let's solve the problem in Figure 1. The first menu of MatMan shows you five options, ranging from entering data in matrices to exiting the program. We want to place values in both matrix X and matrix Y. Press the 1 key, specify a 2×2 matrix, and enter the values you see in Figure 1. Do the same for the Y matrix by pressing the 2 key.

Now ask MatMan to do some computing by selecting option 3. Another menu appears, showing you the operations

you can perform. By pressing the appropriate number key, you can add, subtract, or multiply matrices, multiply matrices by a scalar, or invert or transpose matrices.

First, you'll want to add matrix X to matrix Y (since you should begin, as in all mathematics, with the innermost operation). Press the 1 key to add, then tell the program to add matrix X and matrix Y by entering those two letters. MatMan stores the result in matrix Z. The operation will look like Figure 2.

Figure 2. MatMan Adds

$$\begin{bmatrix} 7 & 17 \\ -20 & 512 \end{bmatrix} + \begin{bmatrix} 32 & 9 \\ 1 & 18 \end{bmatrix} = \begin{bmatrix} 39 & 26 \\ -19 & 530 \end{bmatrix}$$

X Y Z

If you want to check the resulting matrix Z, just press the 4 key when the display returns to the main menu. Matrix Z will appear on the screen. Hit any key to switch back to the main menu.

You're not through yet. Ask MatMan to scalar-multiply matrix Z by the number 3. Press the 3 key to compute again, then the 4 key to select scalar-multiply. Respond with matrix Z and 3 at the appropriate prompts. As before, you can see the result by requesting a *Display result* from the main menu. Notice that MatMan has again stored the result in matrix Z.

Here's the tricky part. You need to multiply what is now in matrix Z by the vector at the far right (2's). No problem. What you need to do is first store the 2's in matrix X, then ask MatMan to compute the product, $Z * X$. Remember that the Z should come first in this case. Figure 3 shows the final answer. Check it by pressing the 4 key when you see the main menu on the screen.

Figure 3. Final Result

$$\begin{bmatrix} 390 \\ 3066 \end{bmatrix}$$

As you can see, MatMan is versatile, fast, and easy to use. It's just what you need when you have complex matrix operations planned.

MatMan

```

100 REM MATMAN
110 REM INITIALIZE
120 GOSUB 190
130 REM CHOOSE FROM MENU
140 GOSUB 640
150 REM MANAGE
160 ON CHOICE GOSUB 820,820,1170,2820
170 IF CHOICE < > 5 THEN 140
180 END
190 REM INITIALIZE
200 PRINT CHR$(21): TEXT : HOME
210 REM GREETING
220 GOSUB 280
230 REM STRINGS
240 GOSUB 400
250 REM MATRIX SIZES
260 GOSUB 480
270 RETURN
280 REM GREETING
290 PRINT "MatMan adds, subtracts, multipl
ies,
300 PRINT "transposes, and inverts any two
310 PRINT "matrices, X and Y.
320 PRINT : PRINT "He stores the result in
Z.
330 PRINT
340 PRINT "Further operations using X, Y,
and
350 PRINT "Z are then allowed.
360 VTAB 23: HTAB 14: PRINT "Press any key
";
370 GET S$
380 HOME
390 RETURN
400 REM STRINGS
410 M$(1) = "Matrix X":M$(2) = "Matrix Y"
420 DATA Add, Subtract, Multiply, Scalar-
multiply, Invert, Transpose, Return to
main menu
430 FOR I = 1 TO 7: READ MT$(I): NEXT
440 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
NEXT I
450 REM DIGITS AFTER DECIMAL
460 DA = 3

```

```

470 RETURN
480 REM MATRIX SIZES
490 DATA 10,10: REM ROWS, COLUMNS
500 READ MR,MC
510 REM DISPLAY
520 PRINT "The maximum sizes of X and Y are
530 VTAB 3: HTAB 14: PRINT "Rows = ";MR
540 VTAB 4: HTAB 11: PRINT "Columns = ";MC

550 PRINT
560 PRINT "Change line 490 for different values.
570 REM DIMENSION
580 M = MC
590 IF MR > MC THEN M = MR
600 DIM X(MR,MC),Y(MR,MC),Z(M,M),F(M,2 * M),S(M,M)
610 VTAB 23: HTAB 14: PRINT "Press any key
";
620 GET S$
630 RETURN
640 REM MENU
650 HOME
660 VTAB 5: HTAB 10: PRINT "Would you like
to
670 INVERSE
680 FOR I = 1 TO 5
690 VTAB 5 + 2 * I: HTAB 11: PRINT I
700 NEXT
710 NORMAL
720 VTAB 7: HTAB 13: PRINT "Enter data on
matrix X
730 VTAB 9: HTAB 13: PRINT "Enter data on
matrix Y
740 VTAB 11: HTAB 13: PRINT "Compute
750 VTAB 13: HTAB 13: PRINT "Display result
760 VTAB 15: HTAB 13: PRINT "Exit
770 VTAB 19: HTAB 10: PRINT "Choice = ? ";
CHR$(7);
780 GET S$
790 CHOICE = VAL (S$)
800 IF CHOICE < 1 OR CHOICE > 5 THEN 770
810 RETURN
820 REM ENTER DATA
830 REM MATRIX SIZE
840 GOSUB 880
850 REM DATA
860 GOSUB 1050

```

```

870 RETURN
880 REM MATRIX SIZES
890 HOME
900 PRINT "Please enter the size of ";M$(C
HOICE);"."
910 VTAB 3: HTAB 21: PRINT SPC( 15); CHR$
(7)
920 VTAB 3: HTAB 14: INPUT "Rows = ";R$
930 R = VAL (R$)
940 IF R < 1 THEN 910
950 IF R > MR THEN VTAB 20: HTAB 4: PRINT
"Sorry, only ";MR;" rows are allowed!"
: GOTO 910
960 VTAB 20: HTAB 4: PRINT SPC( 36)
970 VTAB 5: HTAB 21: PRINT SPC( 15); CHR$
(7)
980 VTAB 5: HTAB 11: INPUT "Columns = ";C$
990 C = VAL (C$)
1000 IF C < 1 THEN 970
1010 IF C > MC THEN VTAB 20: HTAB 2: PRINT
"Sorry, only ";MC;" columns are allowe
d!": GOTO 970
1020 IF CHOICE = 1 THEN RX = R:CY = C
1030 IF CHOICE = 2 THEN RY = R:CY = C
1040 RETURN
1050 REM ENTER DATA
1060 HOME : PRINT CHR$ (7);
1070 PRINT "Please enter data on ";M$(CHOI
CE);"."
1080 FOR I = 1 TO C
1090 VTAB 4: HTAB 1: PRINT "Column No.";I;
":"
1100 FOR J = 1 TO R
1110 VTAB 6: HTAB 4: PRINT "Row No.";J;" =
":COL = POS (K) + 1
1120 PRINT SPC( 15);: VTAB 6: HTAB COL: INPUT
V$
1130 IF CHOICE = 1 THEN X(J,I) = VAL (V$)
1140 IF CHOICE = 2 THEN Y(J,I) = VAL (V$)
1150 NEXT J,I
1160 RETURN
1170 REM COMPUTE
1180 REM PICK
1190 GOSUB 1390
1200 IF PICK = 7 THEN 1380
1210 REM ENTER MATRIX NAME(S)
1220 W$ = "matrix"
1230 IF PICK < 4 THEN W$ = "matrices"
1240 GOSUB 1500

```

Number Crunchers

```
1250 REM LEARN DIMENSIONS
1260 GOSUB 1730
1270 REM CHECK CONFORMABILITY
1280 C$ = "OKAY"
1290 GOSUB 1890
1300 IF C$ < > "OKAY" THEN VTAB 23: HTAB
1: PRINT "Matrix dimensions are noncon-
formable !"; CHR$ (7);: FOR TIME = 1 TO
2500: NEXT TIME: GOTO 1190
1310 REM PLACE MATRICES INTO 'F' AND 'S'
1320 GOSUB 1990
1330 IF PICK < 4 THEN GOSUB 2070
1340 REM COMPUTE
1350 HOME
1360 IF PICK < > 4 THEN VTAB 13: HTAB 15
: FLASH : PRINT "COMPUTING";: FOR TIME
= 1 TO 500: NEXT TIME: NORMAL
1370 ON PICK GOSUB 2150,2150,2230,2320,242
0,2750
1380 RETURN
1390 REM PICK
1400 HOME
1410 VTAB 4: HTAB 12: PRINT "MatMan will:"
: PRINT
1420 FOR I = 1 TO 7
1430 VTAB 2 * I + 4: HTAB 13: INVERSE : PRINT
I;: HTAB 15: NORMAL : PRINT MT$(I)
1440 NEXT
1450 VTAB 20: HTAB 12: PRINT "Choice = ? "
; CHR$ (7);
1460 GET S$
1470 PICK = VAL (S$)
1480 IF PICK < 1 OR PICK > 7 THEN 1450
1490 RETURN
1500 REM NAMES
1510 HOME
1520 PRINT "Please enter the ";W$;" that y
ou
1530 PRINT "would like to ";MT$(PICK);"."
1540 PRINT
1550 PRINT "Matrices: X, Y, or Z
1560 PRINT
1570 REM FIRST
1580 VTAB 6: HTAB 3: PRINT "First = ? "; CHR$
(7);
1590 GET F$
1600 A = ASC (F$)
1610 IF A > 90 THEN A = A - 32
1620 IF A < 88 OR A > 90 THEN 1580
1630 F$ = CHR$ (A): PRINT F$
```

```

1640 IF PICK > 3 THEN 1720
1650 REM SECOND
1660 VTAB 8: HTAB 2: PRINT "Second = ? "; CHR$
    (7);
1670 GET S$
1680 A = ASC (S$)
1690 IF A > 90 THEN A = A - 32
1700 IF A < 88 OR A > 90 THEN 1670
1710 S$ = CHR$ (A): PRINT S$
1720 RETURN
1730 REM LEARN DIMENSIONS
1740 REM FIRST MATRIX
1750 GOSUB 1790
1760 REM SECOND
1770 IF PICK < = 3 THEN GOSUB 1840
1780 RETURN
1790 REM FIRST
1800 IF F$ = "X" THEN RF = RX:CF = CX
1810 IF F$ = "Y" THEN RF = RY:CF = CY
1820 IF F$ = "Z" THEN RF = RZ:CF = CZ
1830 RETURN
1840 REM SECOND
1850 IF S$ = "X" THEN RS = RX:CS = CX
1860 IF S$ = "Y" THEN RS = RY:CS = CY
1870 IF S$ = "Z" THEN RS = RZ:CS = CZ
1880 RETURN
1890 REM CHECK CONFORMABILITY
1900 REM ADDITION & SUBTRACTION
1910 IF (PICK = 1 OR PICK = 2) AND ((RF <
    > RS) OR (CF < > CS)) THEN C$ = "NON
CON"
1920 REM MULTIPLICATION
1930 IF (PICK = 3) AND (CF < > RS) THEN C
$ = "NONCON"
1940 REM INVERSION
1950 IF (PICK = 5) AND (RF < > CF) THEN C
$ = "NONCON"
1960 REM ZERO DIMENSION
1970 IF RF = 0 OR CF = 0 THEN C$ = "NONCON
"
1980 RETURN
1990 REM FIRST
2000 FOR I = 1 TO RF
2010 FOR J = 1 TO CF
2020 IF F$ = "X" THEN F(I,J) = X(I,J)
2030 IF F$ = "Y" THEN F(I,J) = Y(I,J)
2040 IF F$ = "Z" THEN F(I,J) = Z(I,J)
2050 NEXT J,I
2060 RETURN
2070 REM SECOND

```

```

2080 FOR I = 1 TO RS
2090 FOR J = 1 TO CS
2100 IF S$ = "X" THEN S(I,J) = X(I,J)
2110 IF S$ = "Y" THEN S(I,J) = Y(I,J)
2120 IF S$ = "Z" THEN S(I,J) = Z(I,J)
2130 NEXT J,I
2140 RETURN
2150 REM ADD OR SUBTRACT
2160 FOR I = 1 TO RF
2170 FOR J = 1 TO CF
2180 IF PICK = 1 THEN Z(I,J) = F(I,J) + S(
I,J)
2190 IF PICK = 2 THEN Z(I,J) = F(I,J) - S(
I,J)
2200 NEXT J,I
2210 RZ = RF:CZ = CF
2220 RETURN
2230 REM MULTIPLY
2240 FOR I = 1 TO RF
2250 FOR J = 1 TO CS
2260 Z(I,J) = 0
2270 FOR L = 1 TO CF
2280 Z(I,J) = Z(I,J) + F(I,L) * S(L,J)
2290 NEXT L,J,I
2300 RZ = RF:CZ = CS
2310 RETURN
2320 REM SCALAR MULTIPLY
2330 HOME : PRINT CHR$(7);
2340 INPUT "What is the value of your scal
ar ? ";S$
2350 SK = VAL (S$)
2360 FOR I = 1 TO RF
2370 FOR J = 1 TO CF
2380 Z(I,J) = F(I,J) * SK
2390 NEXT J,I
2400 RZ = RF:CZ = CF
2410 RETURN
2420 REM INVERT
2430 GOSUB 2470
2440 REM MOVE MATRIX
2450 GOSUB 2680
2460 RETURN
2470 REM INVERT
2480 REM TACK ON I
2490 FOR I = 1 TO RF
2500 FOR J = 1 TO RF
2510 F(I,RF + J) = 0
2520 IF J = I THEN F(I,RF + J) = 1
2530 NEXT J,I
2540 REM INVERT

```

Number Crunchers

```
2550 FOR I = 1 TO RF
2560 REM ADJUST KEY ROW
2570 V = F(I,I)
2580 FOR J = I TO 2 * RF
2590 F(I,J) = F(I,J) / V
2600 NEXT J
2610 REM ADJUST REMAINING ROWS
2620 FOR J = 1 TO RF
2630 X = F(J,I)
2640 FOR L = I TO 2 * RF
2650 IF J < I THEN F(J,L) = F(J,L) - X *
    F(I,L)
2660 NEXT L,J,I
2670 RETURN
2680 REM MOVE MATRIX
2690 FOR I = 1 TO RF
2700 FOR J = 1 TO CF
2710 Z(I,J) = F(I,RF + J)
2720 NEXT J,I
2730 RZ = RF:CZ = CF
2740 RETURN
2750 REM TRANSPOSE
2760 FOR I = 1 TO CF
2770 FOR J = 1 TO RF
2780 Z(I,J) = F(J,I)
2790 NEXT J,I
2800 RZ = CF:CZ = RF
2810 RETURN
2820 REM DISPLAY RESULTS
2830 R = RZ:C = CZ
2840 FOR Q = 1 TO R STEP 10
2850 FOR I = 1 TO C STEP 2
2860 REM HEADING
2870 GOSUB 2920
2880 REM BODY
2890 GOSUB 3140
2900 NEXT I,Q
2910 RETURN
2920 REM HEADING
2930 HOME
2940 PRINT L$
2950 PRINT TAB( 16)"MATRIX Z
2960 PRINT L$
2970 REM COLUMN HEADING
2980 COL = 14
2990 FOR L = I TO I + 1
3000 IF L <= C THEN VTAB 5: HTAB COL: PRINT
    "Column ";L
3010 COL = COL + 16
3020 NEXT L
```


Number Crunchers

```
3030 REM BRACKETS
3040 INVERSE
3050 FOR M = 7 TO 18 STEP 11
3060 VTAB M: HTAB 7: PRINT CHR$ (32);: HTAB
(38): PRINT CHR$ (32)
3070 NEXT M
3080 FOR M = 7 TO 18
3090 VTAB M: HTAB 6: PRINT CHR$ (32)
3100 VTAB M: HTAB 39: PRINT CHR$ (32)
3110 NEXT M
3120 NORMAL
3130 RETURN
3140 REM VALUES
3150 ROW = 8
3160 FOR J = 0 TO 9
3170 IF J > R THEN 3250
3180 VTAB ROW: HTAB 1: PRINT "R ";J;
3190 COL = 21
3200 FOR L = I TO I + 1
3210 IF L < = C THEN VTAB ROW:HT = COL:N
R = Z(J,L): GOSUB 9000
3220 COL = COL + 16
3230 NEXT L
3240 ROW = ROW + 1
3250 NEXT J
3260 VTAB 22: HTAB 1: PRINT L$
3270 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
3280 GET S$
3290 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "- "
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN
```



4 Simple Statistics



Simple Statistics

We see an incredible quantity of numerical information each day, everything from batting averages to stock market prices to monthly utility bills. Here are a few simple programs that will help you transform your raw data into a useful, understandable form.

Interpreting data is important to all of us. Sometimes it's difficult to understand and use numbers. But once they're translated into a form that we can comprehend, they can show trends, percentages, and frequencies. These nine programs will help you understand the raw numbers.

- **Mean, Variance, and Standard Deviation.** Compute the average value of a set of observations or calculate how diverse the observations are.
- **Relative and Cumulative Frequencies.** Figure the percentage of total observations that fall into each of several different categories.
- **Pie Plot.** Draw circular charts with up to nine segments.
- **Frequency Plot.** Draw bar graphs with up to ten percentages.
- **Analysis of Variance.** Test various population means to see if they're equal.
- **Simple Correlation Coefficients.** Estimate the linear association between a group of variables.
- **Simple Linear Regression Analysis.** Fit a trend line through a group of observations.
- **Multiple Linear Regression Analysis.** Estimate the linear relationship between one variable and several other variables.
- **General-Form Curve Fitter.** Use in estimating any one of four different types of simple regression equations: linear, power, exponential, and reciprocal.

A Note on Numbers

Anytime you work with measurement and numbers, you should keep in mind the possible sources of error. Today's home computers are generally very accurate, but are on occasion subject to rounding errors. Use common sense when

interpreting the results obtained from these programs. If the answer to a derivative calculation is 1000.0001, you should consider rounding this to 1000 if you use it in another calculation. The difference is most likely insignificant, since it is smaller than the possible error in the calculation.

Mean, Variance, and Standard Deviation

A list of numbers can be turned into something meaningful when you compute such things as mean, variance, and standard deviation. With this program, you can turn raw test scores or even yards gained by a running back into something understandable.

The fullback pounds through the line for a gain of 4.3 yards. Behind powerful blocking, he carries again on second down for 3.7 more yards. And on third and two he picks up the crucial first down. Over the course of a late game-winning drive, the fullback carries the ball 12 times and gains the yardage shown in Table 1. Pretty impressive. But you may want to know how many yards he gained per carry.

Table 1. Fullback's Yards Gained

Carry	Yards
1	4.3
2	3.7
3	2.0
4	5.0
5	4.1
6	3.8
7	3.9
8	6.8
9	-1.2
10	6.5
11	4.5
12	4.4

Means

The computer can help you find that out by computing the mean, variance, and standard deviation of a group of observations. Simply type in this short program and run it.

The *mean* is the average value of a series, or the sum of all values divided by the number of observations. To compute

average yards per carry, for instance, just enter the number of yards that the fullback gained in each run. Type in 12 to indicate the number of observations (runs in our example), then each value. Use the numbers in Table 1. The computer adds these values, divides by 12, and displays the answer of 3.983. That's the average number of yards the fullback carried the ball each time he ran.

Variance and Standard Deviation

You'll also see two other numbers listed. *Variance* measures the dispersion of observations about the mean. The higher the variance, the greater, or wider, the spread. Notice that the fullback's variance is 4.252. The *standard deviation*, which is 2.062 in this example, is simply the square root of the variance.

A variance in itself is meaningless. You need to evaluate it in relationship to another variance. Everything, after all, is relative. Hence, our example variance takes on meaning only when compared to a similar figure, that is, to the variance of yards gained by another player.

Table 2 shows the number of yards gained in each carry by the halfback.

Table 2. Halfback's Yards Gained

Carry	Yards
1	2.0
2	1.3
3	17.0
4	-3.6
5	-2.0
6	10.2
7	2.4
8	11.8
9	2.3
10	0.5

Using the program still in memory, enter these ten values. After a moment of calculation, the computer displays the halfback's mean, variance, and standard deviation. He gained an average of 4.190 yards per carry, close to the fullback's average of 3.983. However, his variance is a huge 43.408. Compare this to the fullback's variance of 4.252. This means that

the halfback's performance is more erratic than the fullback's. The former may gain few if any yards on any one play and then break a 40-yard run the next.

Comparing lists of numbers is easy with "Mean, Variance, and Standard Deviation."

Mean, Variance, and Standard Deviation

```
100 REM MEAN, VARIANCE, & STANDARD DEVIAT
    ION
110 REM INITIALIZE
120 GOSUB 200
130 REM ENTER DATA
140 GOSUB 340
150 REM COMPUTE
160 GOSUB 590
170 REM DISPLAY
180 GOSUB 730
190 END
200 REM INITIALIZE
210 PRINT CHR$(21): TEXT : HOME
220 REM MAXIMUM NUMBER OF OBSERVATIONS
230 DATA 300
240 READ NX
250 DIM X(NX)
260 REM HEADING
270 PRINT " This program computes the mea
    n,
280 PRINT "variance, and standard deviatio
    n of a
290 PRINT "group of observations.
300 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT
310 REM DIGITS AFTER DECIMAL
320 DA = 3
330 RETURN
340 REM DATA
350 REM NUMBER OF OBSERVATIONS
360 GOSUB 400
370 REM VALUES
380 GOSUB 490
390 RETURN
400 REM NUMBER OF OBSERVATIONS
410 VTAB 5: HTAB 1: PRINT "How many observ
    ations do you
420 VTAB 6: HTAB 8: PRINT SPC(15); CHR$(
    7)
430 VTAB 6: HTAB 1: INPUT "have ? ";N$
440 N = INT ( VAL (N$))
```

```
450 VTAB 22: HTAB 1: PRINT SPC( 40)
460 IF N < 2 THEN VTAB 22: HTAB 10: PRINT
    "I need at least two !": GOTO 420
470 IF N > NX THEN VTAB 22: HTAB 9: PRINT
    "Only ";NX;" are allowed !": GOTO 420
480 RETURN
490 REM VALUES
500 HOME
510 PRINT "Please enter your data.
520 FOR J = 1 TO N
530 VTAB 3: HTAB 17: PRINT SPC( 15)
540 VTAB 3: HTAB 1: PRINT "Value No. ";J; TAB(
    14);"=" ";
550 INPUT X$
560 X(J) = VAL (X$)
570 NEXT
580 RETURN
590 REM COMPUTE
600 HOME
610 VTAB 12: HTAB 15: FLASH : PRINT "COMPU
    TING": NORMAL
620 REM KEY TERMS
630 S = 0: SQ = 0
640 FOR I = 1 TO N
650 S = S + X(I)
660 SQ = SQ + X(I) ^ 2
670 NEXT
680 REM DESIRED VALUES
690 MEAN = S / N
700 VARIANCE = (SQ - S ^ 2 / N) / (N - 1)
710 STDEV = SQR (VARIANCE)
720 RETURN
730 REM DISPLAY
740 HOME
750 PRINT L$
760 PRINT TAB( 14)"KEY STATISTICS
770 PRINT L$
780 VTAB 6: HTAB 7: PRINT "Mean"; TAB( 17)
    "="";:HT = 32:NR = MEAN: GOSUB 9000
790 VTAB 8: HTAB 7: PRINT "Variance"; TAB(
    17)"="";:NR = VARIANCE: GOSUB 9000
800 VTAB 10: HTAB 7: PRINT "Standard
810 PRINT TAB( 7)"Deviation ="";:NR = STDE
    V: GOSUB 9000
820 VTAB 20: HTAB 1: PRINT L$
830 VTAB 23: HTAB 14: PRINT "Press any key
    ";
840 GET S$
850 RETURN
```

```

9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Relative and Cumulative Frequencies

Seeing the percentage breakdown of a large number of observations, whether they're items in a budget or days when sales quotas are met, can be illuminating. Use this short program to calculate relative and cumulative percentages.

Figuring percentages with pencil and paper, or even a hand calculator, can take time, especially if you have a large list of numbers. When you use "Relative and Cumulative Frequencies," however, all you have to do is enter the values. Your computer shows you the rest.

Let's take a typical household's budget as an example. The table shows the categories and the amount spent for each.

Spending

Category	Amount
Movies	\$ 50
Clothes For Teenagers	175
Allowance	40
Hairdresser	125
Dog Food	175
Mortgage Payments	950
Transportation	70
Utilities	110
Food	300
Total	\$1,995

The total for each month is \$1,995, hardly a budget that allows much saving. The family might even be overextending itself. To decide what can be trimmed in the budget, it would be valuable to know each category's percentage of the total. After all, the family's teenagers may not believe they're part of the problem. With this program it's simple to find out.

Class and Frequency

Once you've entered and saved Relative and Cumulative Frequencies, run it. You're first asked to enter the number of *classes*, or categories of information. Key in 9 to match the number of spending categories in the table.

Next, enter the *frequency* of each class, or the number of observations. In the example, number of dollars would be the frequency. If you were calculating the proportion of employees' sick days to work days, it would be the number of days taken off for illness. Enter 50 for the first class (movies), 175 for the second class (clothes for teenagers), 40 for the third class (allowance), and so on. The computer then displays the percentages shown in the figure, along with the actual amount entered for frequency.

Percentages

Category	Relative Frequency	Cumulative Frequency
Movies	2.51	2.51
Clothes For Teenagers	8.77	11.28
Allowance	2.01	13.28
Hairdresser	6.27	19.55
Dog Food	8.77	28.32
Mortgage Payments	47.62	75.94
Transportation	3.51	79.45
Utilities	5.51	84.96
Food	15.04	100.00

Each *relative* frequency indicates what percentage of the total amount was spent in each category. Only 2.51 percent went to movies, but a surprising 6.27 percent was spent at the hairdresser. And surprisingly, the dog's food was close to 9 percent of total monthly spending.

The family could cut down somewhere. What about the first five items (movies through dog food)? Looking at the screen again, our imaginary family can see that 28.32 percent of total spending was for these items. This figure is called a *cumulative* frequency. It's computed by summing the amount

spent on the first five items, and then dividing by total spending ($\$565/\$1995 = 0.2832 = 28.32\%$). The other cumulative frequencies are calculated in the same way.

By organizing and inputting the numbers in a different order, you could see different cumulative frequencies. The flexibility of Relative and Cumulative Frequencies makes it easy to do.

Relative and Cumulative Frequencies

```
100 REM RELATIVE & CUMULATIVE FREQUENCIES
110 REM INITIALIZE
120 GOSUB 200
130 REM ENTER DATA
140 GOSUB 420
150 REM COMPUTE
160 GOSUB 740
170 REM DISPLAY RESULTS
180 GOSUB 940
190 END
200 REM INITIALIZE
210 PRINT CHR$(21): TEXT : HOME
220 REM MAXIMUM NUMBER OF CLASSES
230 DATA 200
240 READ NX
250 DIM F(NX), RF(NX), CF(NX)
260 REM HEADING
270 GOSUB 290
280 RETURN
290 REM HEADING
300 PRINT " This program computes relative
    e and
310 PRINT "cumulative frequencies.
320 PRINT
330 PRINT "A frequency is the number of ti
    mes that
340 PRINT "an observation occurs.
350 PRINT
360 PRINT "And a relative frequency is the
370 PRINT "percentage of times that it occ
    urs.
380 L$ = "": FOR I = 1 TO 39: L$ = L$ + " ":
    NEXT
390 VTAB 23: HTAB 14: PRINT "Press any key
    ";
400 GET S$
410 RETURN
420 REM ENTER DATA
430 REM CLASSES
440 GOSUB 480
```

```
450 REM FREQUENCIES
460 GOSUB 570
470 RETURN
480 REM CLASSES
490 HOME
500 VTAB 1: HTAB 31: PRINT SPC( 9); CHR$
    (7)
510 VTAB 1: HTAB 1: INPUT "How many classe
    s do you have ? ";C$
520 N = INT ( VAL (C$))
530 VTAB 23: HTAB 9: PRINT SPC( 31)
540 IF N < 1 THEN VTAB 23: HTAB 10: PRINT
    "I need at least one !": GOTO 500
550 IF N > NX THEN VTAB 23: HTAB 9: PRINT
    "Only ";NX;" are allowed !": GOTO 500
560 RETURN
570 REM FREQUENCIES
580 HOME
590 PRINT "Please enter the frequencies fo
    r each
600 PRINT "class."; CHR$ (7)
610 SUM = 0
620 FOR I = 1 TO N
630 VTAB 4: HTAB 17: PRINT SPC( 20)
640 VTAB 4: HTAB 1: PRINT "Class No. ";I; TAB(
    14)"= ";
650 INPUT V$
660 V = VAL (V$): IF V < 0 THEN PRINT CHR$
    (7): GOTO 630
670 F(I) = V: SUM = SUM + V
680 NEXT
690 IF SUM > 0 THEN 730
700 VTAB 20: HTAB 8: PRINT "Sum is zero !
    Try again."
710 VTAB 23: HTAB 14: PRINT "Press any key
    ";
720 GET S$: GOTO 580
730 RETURN
740 REM COMPUTE
750 HOME
760 VTAB 12: HTAB 15: FLASH : PRINT "COMPU
    TING": NORMAL
770 FOR PAUSE = 1 TO 500: NEXT
820 CF(0) = 0
830 FOR I = 1 TO N
840 RF(I) = F(I) * 100 / SUM
845 CF(I) = CF(I - 1) + RF(I)
850 NEXT
860 RETURN
940 REM DISPLAY RESULTS
```

```

950 FOR J = 1 TO N STEP 10
960 REM HEADING
970 GOSUB 1020
980 REM BODY
990 GOSUB 1090
1000 NEXT J
1010 RETURN
1020 REM HEADING
1030 HOME
1040 PRINT L$
1050 PRINT TAB( 15)"FREQUENCIES
1060 PRINT L$
1070 VTAB 5: HTAB 1: PRINT "Class"; TAB( 1
2)"Actual"; TAB( 20)"Relative"; TAB( 3
0)"Cumulative
1080 RETURN
1090 REM BODY
1100 R = 7
1110 FOR L = J TO J + 9
1120 IF L > N THEN 1200
1130 VTAB R:DA = 0:HT = 3:NR = L: GOSUB 90
00
1140 VTAB R:DA = 2:HT = 17:NR = F(L): GOSUB
9000
1150 VTAB R:HT = 26:NR = RF(L): GOSUB 9000

1160 VTAB R: HTAB 27: PRINT "%"
1170 VTAB R:HT = 38:NR = CF(L): GOSUB 9000

1180 VTAB R: HTAB 39: PRINT "%"
1190 R = R + 1
1200 NEXT L
1210 VTAB 22: HTAB 1: PRINT L$
1220 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
1230 GET S$
1240 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)

```



```
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3, 1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN
```

Pie Plot

A pie chart can graphically show how a whole is divided into parts. After entering values for up to nine "slices," this program quickly draws a chart and labels each piece.

Pie charts are often used to illustrate percentages of a total. Whether you're dealing with components of a budget or total sales separated by salesperson, such a chart can quickly show you the parts of a whole. This program plots and draws a pie chart of up to nine slices. You can even label them yourself and see each slice's percentage of the total. All you have to do is enter the values.

Pieces of the Pie

Type in, save, and run the program. The computer does most of the work for you. All you have to supply are the number of slices, the titles you want associated with each slice, and the value for each slice. The program computes the percentages.

Assume, for instance, that you want to compute and plot percentage sales for each of four employees of C. D. Bugs, a local insect exterminator. The table shows each salesperson's basic data.

Yearly Sales

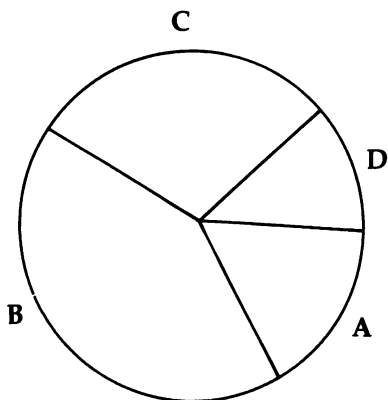
Employee	Sales
Al Borden	\$25,000
John Kowalski	50,000
Ruth Rhone	47,000
Babs Nixon	12,000

Since the example has four entries, press the 4 key and RETURN. Then type in the title of each slice, in this case the last names of the salespeople in the table. Up to eight characters can be entered for each title. Next, enter the raw data for each slice. Those are the sales figures in the right-hand column of the table. Type in 25, 50, 47, and 12 (you don't need to type in the zeros indicating thousands). The computer calculates the percentages for you. Finally, key in a title for the

pie chart, something like "Percentage Sales at C. D. Bugs," for instance.

The pie chart shows in the center of the screen, each slice labeled. At the bottom of the screen will be the prompt, *Analyze a slice (Y, N)?* Press Y and the letter of the slice you want to analyze.

Pie Chart



Percentage of Sales at C. D. Bugs

Analyze a slice (Y/N)?

Any slice less than 2.5 percent of the total will be drawn but not labeled, because of the lack of room. The percentage, however, may still be analyzed.

The program faithfully represents percentages, but has difficulty representing vertical lines on the screen. As a result, occasionally two adjacent sections will not have a divider.

Technical note. Line 370 contains the aspect ratio of your video screen. This ratio determines how circular the pie chart is drawn. To determine the aspect ratio, type in and run the following program.

```
10 HOME: HGR: HCOLOR=3
20 HPLOT 0,0 TO 150,0
30 HPLOT 0,0 TO 0,150
40 GETS$: TEXT
```

Measure the two lines, and divide the length of the horizontal line by the length of the vertical line. The result is the aspect ratio. Use this value in line 370 of "Pie Plot."

Pie Plot

```
100 REM   PIE PLOT
110 REM   INITIALIZE
120 GOSUB 210
130 REM   ENTER DATA
140 GOSUB 490
150 REM   COMPUTE
160 GOSUB 1010
170 REM   PLOT
180 GOSUB 1210
190 TEXT : HOME : PRINT "Bye-bye
200 END
210 REM   INITIALIZE
220 PRINT CHR$(21): TEXT : HOME
230 REM   TITLE
240 GOSUB 280
250 REM   GREETING
260 GOSUB 320
270 RETURN
280 REM   TITLE
290 VTAB 12: HTAB 16: PRINT "Pie Plot
300 FOR PAUSE = 1 TO 1500: NEXT
310 RETURN
320 REM   GREETING
330 HOME
340 PI = 3.14159
350 DT = PI / 200
360 RADIUS = 55
370 AR = 1: REM   ASPECT RATIO (LENGTH OF V
    ERTICAL TO HORIZONTAL AXIS)
380 DEF FN X(I) = 140 + RADIUS * COS (I)
390 DEF FN Y(I) = 80 + RADIUS * AR * SIN
    (I)
400 REM   MAXIMUM NUMBER OF SLICES
410 DATA 9
420 READ NX
430 DIM X(NX),NM$(NX),PCT(NX),RAD(NX),CRAD
    (NX)
440 PRINT "   This program plots a pie for
    you. Up
450 PRINT "to ";NX;" slices are allowed, w
    ith each
460 PRINT "piece representing its percenta
    ge of
470 PRINT "the total.
480 RETURN
490 REM   ENTER DATA
500 REM   NUMBER OF SLICES
510 GOSUB 590
```

```
520 REM NAME OF EACH SLICE
530 GOSUB 650
540 REM VALUES
550 GOSUB 770
560 REM TITLE
570 GOSUB 890
580 RETURN
590 REM NUMBER OF SLICES
600 VTAB 7: HTAB 31: PRINT SPC( 9); CHR$
    (7)
610 VTAB 7: HTAB 1: INPUT "How many slices
    do you have ? ";N$
620 N = INT ( VAL (N$))
630 IF N < 1 OR N > NX THEN 600
640 RETURN
650 REM NAMES
660 HOME
670 PRINT "Please name each slice. Up to
    15
680 PRINT "characters will be shown."; CHR$
    (7)
690 FOR I = 1 TO N
700 VTAB 5: HTAB 15: PRINT SPC( 25)
710 VTAB 5: HTAB 1: PRINT "Name No. ";I; TAB(
    12)"=" ";
720 INPUT NM$(I)
730 IF NM$(I) = "" THEN PRINT CHR$ (7): GOTO
    700
740 IF LEN (NM$(I)) > 15 THEN NM$(I) = LEFT$
    (NM$(I),15)
750 NEXT
760 RETURN
770 REM VALUES
780 HOME
790 PRINT "Please enter values for each sl
    ice.
800 PRINT "I'll compute the percentages.";
    CHR$ (7)
810 FOR I = 1 TO N
820 VTAB 5: HTAB 15: PRINT SPC( 20)
830 VTAB 5: HTAB 1: PRINT "Slice No. ";I; TAB(
    13)"=" ";
840 INPUT V$
850 X(I) = VAL (V$)
860 IF X(I) < = 0 THEN 820
870 NEXT
880 RETURN
890 REM TITLE
900 HOME
```

```
910 PRINT "Please title your plot. Up to
40
920 PRINT "characters will be displayed.
930 VTAB 5: HTAB 1: INPUT "Title = ? ";TITLE$
940 IF LEN (TITLE$) > 40 THEN TITLE$ = LEFT$
(TITLE$,40)
950 VTAB 9: HTAB 1: PRINT "Are you using a
color monitor (Y/N) ? "; CHR$ (7);
960 GET S$
970 IF S$ < > "Y" AND S$ < > "y" AND S$ <
> "N" AND S$ < > "n" THEN 950
980 C1 = 0:C2 = 3:C3 = 0
990 IF S$ = "Y" OR S$ = "y" THEN C1 = 6:C2
= 5:C3 = 4
1000 RETURN
1010 REM COMPUTE
1020 REM TOTAL
1030 SUM = 0
1040 FOR I = 1 TO N
1050 SUM = SUM + X(I)
1060 NEXT
1070 REM PERCENTAGES
1080 FOR I = 1 TO N
1090 PCT(I) = X(I) / SUM
1100 NEXT
1110 REM EACH ANGLE IN RADIANS
1120 FOR I = 1 TO N
1130 RAD(I) = 2 * PI * PCT(I)
1140 NEXT
1150 REM CUMULATIVE ANGLES IN RADIANS
1160 CRAD(0) = 0
1170 FOR I = 1 TO N
1180 CRAD(I) = RAD(I) + CRAD(I - 1)
1190 NEXT
1200 RETURN
1210 REM PLOT
1220 HOME
1230 HGR : HCOLOR= C1: HPLOT 1,1: CALL 624
54
1240 REM BOX
1250 GOSUB 1370
1260 REM CIRCLE
1270 GOSUB 1430
1280 REM SLICES
1290 GOSUB 1480
1300 REM LABEL
1310 GOSUB 1560
1320 REM TITLE
```

```
1330 GOSUB 1670
1340 REM ANALYZE
1350 GOSUB 1710
1360 RETURN
1370 REM BOX
1380 HCOLOR= C3: FOR I = 157 TO 159: HPLLOT
    0,I TO 279,I: NEXT
1390 HCOLOR= C2
1400 HPLLOT 0,0 TO 279,0: HPLLOT TO 279,156
    : HPLLOT TO 0,156: HPLLOT TO 0,0
1410 HPLLOT 1,1 TO 278,1: HPLLOT TO 278,155
    : HPLLOT TO 1,155: HPLLOT TO 1,1
1420 RETURN
1430 REM CIRCLE
1440 FOR I = 0 TO 2 * PI STEP DT
1450 HPLLOT 140,80 TO FN X(I), FN Y(I)
1460 NEXT
1470 RETURN
1480 REM SLICES
1490 HCOLOR= C3
1500 FOR I = 1 TO N
1510 X = FN X(CRAD(I))
1520 Y = FN Y(CRAD(I))
1530 HPLLOT 140,80 TO X,Y
1540 NEXT
1550 RETURN
1560 REM LABEL
1570 FOR I = 1 TO N
1580 IF PCT(I) < .025 THEN 1650
1590 V = CRAD(I - 1) + RAD(I) / 2
1600 X = 140 + (RADIUS + 12) * COS (V)
1610 Y = 80 + (RADIUS + 12) * AR * SIN (V)
1620 GOSUB 2000
1630 HCOLOR= 3
1640 ON I GOSUB 2060,2100,2150,2190,2230,2
    270,2300,2350,2390
1650 NEXT I
1660 RETURN
1670 REM TITLE
1680 L = LEN (TITLE$)
1690 VTAB 21: HTAB (21 - L / 2): PRINT TIT
    LE$
1700 RETURN
1710 REM ANALYZE
1720 VTAB 23: HTAB 1: PRINT SPC( 40): VTAB
    24: HTAB 14: PRINT SPC( 14)
1730 VTAB 23: HTAB 1: PRINT "Analyze a sli
    ce (Y/N) ? "; CHR$ (7);
1740 GET S$
1750 IF S$ = "N" OR S$ = "n" THEN 1840
```

```
1760 IF S$ < > "Y" AND S$ < > "y" THEN 1
730
1770 REM CHOOSE
1780 GOSUB 1850
1790 REM DISPLAY PERCENTAGE
1800 GOSUB 1940
1810 VTAB 24: HTAB 14: INVERSE : PRINT "PRESS ANY KEY ";: NORMAL
1820 GET S$
1830 GOTO 1720
1840 RETURN
1850 REM CHOOSE
1860 VTAB 23: HTAB 1: PRINT SPC( 24)
1870 VTAB 23: HTAB 1: PRINT "Which one ? "
; CHR$ (7);
1880 GET S$
1890 A = ASC (S$)
1900 IF A > 90 THEN A = A - 32
1910 NM = A - 64
1920 IF NM < 1 OR NM > N THEN 1870
1930 RETURN
1940 REM PERCENTAGE
1950 V = PCT(NM) * 100
1960 P = INT (V * 10 + .5) / 10
1970 VTAB 23: HTAB 1: PRINT SPC( 12)
1980 VTAB 23: HTAB 1: PRINT "Slice "; CHR$
(A);": ";NM$(NM);" = ";P;" %"
1990 RETURN
2000 REM BLACK BACKGROUND FOR LETTER
2010 HCOLOR= 0:X1 = X - 1:Y1 = Y + 1
2020 FOR L = X1 TO X1 + 6
2030 HPLLOT L,Y1 TO L,Y1 - 8
2040 NEXT L
2050 RETURN
2060 REM A
2070 HPLLOT X,Y TO X,Y - 4: HPLLOT X + 4,Y TO
X + 4,Y - 4
2080 HPLLOT X,Y - 2 TO X + 4,Y - 2: HPLLOT X
+ 1,Y - 5: HPLLOT X + 3,Y - 5: HPLLOT X
+ 2,Y - 6
2090 RETURN
2100 REM B
2110 HPLLOT X,Y TO X,Y - 6: HPLLOT X,Y - 6 TO
X + 3,Y - 6
2120 HPLLOT X,Y TO X + 3,Y: HPLLOT X,Y - 3 TO
X + 3,Y - 3
2130 HPLLOT X + 4,Y - 5: HPLLOT X + 4,Y - 4:
HPLLOT X + 4,Y - 1: HPLLOT X + 4,Y - 2
2140 RETURN
```



```
2150 REM C
2160 HPLOT X,Y - 1 TO X,Y - 5: HPLOT X + 1
    ,Y - 6 TO X + 3,Y - 6
2170 HPLOT X + 1,Y TO X + 3,Y: HPLOT X + 4
    ,Y - 5: HPLOT X + 4,Y - 1
2180 RETURN
2190 REM D
2200 HPLOT X,Y TO X,Y - 6: HPLOT X,Y TO X +
    3,Y
2210 HPLOT X,Y - 6 TO X + 3,Y - 6: HPLOT X
    + 4,Y - 1 TO X + 4,Y - 5
2220 RETURN
2230 REM E
2240 HPLOT X + 4,Y TO X,Y: HPLOT TO X,Y -
    6
2250 HPLOT TO X + 4,Y - 6: HPLOT X,Y - 3 TO
    X + 3,Y - 3
2260 RETURN
2270 REM F
2280 HPLOT X,Y TO X,Y - 6: HPLOT TO X + 4
    ,Y - 6: HPLOT X,Y - 3 TO X + 3,Y - 3
2290 RETURN
2300 REM G
2310 HPLOT X,Y - 1 TO X,Y - 5: HPLOT X + 1
    ,Y - 6 TO X + 4,Y - 6
2320 HPLOT X + 1,Y TO X + 4,Y: HPLOT X + 4
    ,Y - 1
2330 HPLOT X + 4,Y - 2: HPLOT X + 3,Y - 2
2340 RETURN
2350 REM H
2360 HPLOT X,Y TO X,Y - 6: HPLOT X + 4,Y TO
    X + 4,Y - 6
2370 HPLOT X,Y - 3 TO X + 4,Y - 3
2380 RETURN
2390 REM I
2400 HPLOT X + 2,Y TO X + 2,Y - 6
2410 HPLOT X + 1,Y TO X + 3,Y: HPLOT X + 1
    ,Y - 6 TO X + 3,Y - 6
2420 RETURN
```

Frequency Plot

Graphs let you see at a glance what otherwise might be lost in a long column of figures. Indeed, if a picture is worth a thousand words, then a good graph is worth a thousand numbers. This program, which illustrates percentages in the form of a bar graph, is no exception.

“Frequency Plot” works in much the same way as another program in this chapter, “Relative and Cumulative Frequencies.” Instead of simply listing the relative percentages, however, this program draws a bar chart. You can get an idea of each item’s percentage of the total much faster.

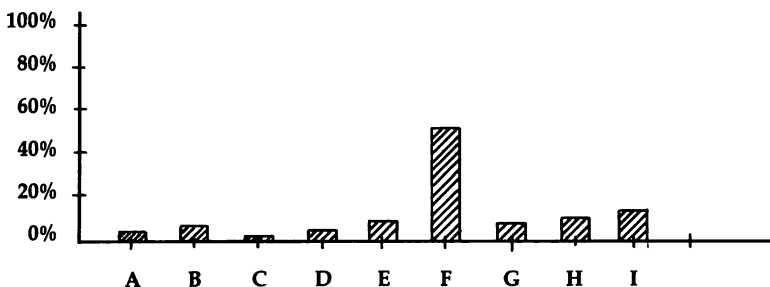
As in Relative and Cumulative Frequencies, you need to enter the number of classes and the frequencies’ values. The computer does the rest.

Bar Frequency

Enter and save Frequency Plot, then run it. Type in the number of classes, then the frequency of each class. As an example, use the values from the table found in the article explaining the Relative and Cumulative Frequencies program. You will also need to indicate whether or not you have a color monitor.

You’ll then see a bar graph charting the relative frequency (percentages) of each class, or category. If you use the data from the table, you’ll see something like the figure below on the screen. Notice that class 6, mortgage payments, towers over all other expenses. The ninth class, food, is the second highest bar.

Budget Charting



In short, this frequency plotting program lets you quickly and easily visualize how the family, or the business, spends its money. Important differences in expenditures are emphasized by different-size columns, with the height of each drawn according to its percentage of total outlays. Of course, there are many other uses for the Frequency Plot program.

Frequency Plot

```
100 REM RELATIVE FREQUENCY PLOT
110 REM INITIALIZE
120 GOSUB 190
130 REM ENTER DATA
140 GOSUB 320
150 REM PLOT
160 GOSUB 780
170 TEXT : HOME : PRINT "Bye-bye
180 END
190 REM INITIALIZE
200 PRINT CHR$(21): TEXT : HOME
210 PRINT " This program plots relative
220 PRINT "frequencies.
230 PRINT
240 PRINT "A frequency is the number of ti
mes that
250 PRINT "an observation occurs.
260 PRINT
270 PRINT "And a relative frequency is the
280 PRINT "percentage of times that it occ
urs.
290 VTAB 23: HTAB 14: PRINT "Press any key
";
300 GET S$
310 RETURN
320 REM ENTER DATA
330 REM CLASSES
340 GOSUB 400
350 REM OBSERVATIONS
360 GOSUB 490
370 REM TITLE
380 GOSUB 660
390 RETURN
400 REM CLASSES
410 HOME
420 VTAB 1: HTAB 31: PRINT SPC( 9); CHR$(
(7)
430 VTAB 1: HTAB 1: INPUT "How many classe
s do you have ? ";K$
440 K = INT ( VAL (K$))
```

```
450 VTAB 23: HTAB 9: PRINT SPC( 31)
460 IF K < 1 THEN VTAB 23: HTAB 10: PRINT
    "I need at least one !": GOTO 420
470 IF K > 10 THEN VTAB 23: HTAB 9: PRINT
    "Only ten are allowed !": GOTO 420
480 RETURN
490 REM OBSERVATIONS
500 HOME
510 PRINT "Please enter the frequencies fo
    r each
520 PRINT "class."; CHR$ (7)
530 SUM = 0
540 FOR I = 1 TO K
550 VTAB 4: HTAB 16: PRINT SPC( 20)
560 VTAB 4: HTAB 1: PRINT "Class No. ";I; TAB(
    13)"= ";
570 INPUT V$
580 V = VAL (V$): IF V < 0 THEN PRINT CHR$
    (7): GOTO 550
590 CLASS(I) = V:SUM = SUM + V
600 NEXT
610 IF SUM > 0 THEN 650
620 VTAB 20: HTAB 8: PRINT "Sum is zero !
    Try again."
630 VTAB 23: HTAB 14: PRINT "Press any key
    ";
640 GET S$: GOTO 500
650 RETURN
660 REM TITLE
670 HOME
680 PRINT "Please title your plot. Up to
    40
690 PRINT "characters will be displayed.";
    CHR$ (7)
700 VTAB 5: HTAB 1: INPUT "Title = ? ";TIT
    LE$
710 IF LEN (TITLE$) > 40 THEN TITLE$ = LEFT$
    (TITLE$,40)
720 VTAB 9: HTAB 1: PRINT "Are you using a
    color monitor (Y/N) ? "; CHR$ (7);
730 GET S$
740 IF S$ < > "Y" AND S$ < > "y" AND S$ <
    > "N" AND S$ < > "n" THEN 720
750 C1 = 0:C2 = 3
760 IF S$ = "Y" OR S$ = "y" THEN C1 = 1:C2
    = 2
770 RETURN
780 REM PLOT
790 HOME
800 HGR : HCOLOR= C1: HPLLOT 1,1: CALL 6245
    4
```

```
810 REM BOX
820 GOSUB 940
830 REM AXES
840 GOSUB 1000
850 REM LABEL
860 GOSUB 1110
870 REM BARS
880 GOSUB 1460
890 REM TITLE
900 GOSUB 1710
910 REM ANALYZE
920 GOSUB 1750
930 RETURN
940 REM BOX
950 HCOLOR= 0: FOR I = 157 TO 159: HPLOT 0
    ,I TO 279,I: NEXT
960 HCOLOR= C2
970 HPLOT 0,0 TO 279,0: HPLOT TO 279,156:
    HPLOT TO 0,156: HPLOT TO 0,0
980 HPLOT 1,1 TO 278,1: HPLOT TO 278,155:
    HPLOT TO 1,155: HPLOT TO 1,1
990 RETURN
1000 REM AXES
1010 HPLOT 50,30 TO 50,130: HPLOT TO 260,
    130
1020 REM Y-AXIS TIC MARKS
1030 FOR I = 30 TO 130 STEP 20
1040 HPLOT 48,I TO 52,I
1050 NEXT
1060 REM X-AXIS TIC MARKS
1070 FOR I = 50 TO 250 STEP 20
1080 HPLOT I,128 TO I,132
1090 NEXT
1100 RETURN
1110 REM LABEL
1120 REM Y-AXIS
1130 GOSUB 1170
1140 REM X-AXIS
1150 GOSUB 1390
1160 RETURN
1170 REM Y-AXIS
1180 REM BLOCK-OUT AREA
1190 FOR Y = 34 TO 134 STEP 20
1200 FOR X = 18 TO 36 STEP 6
1210 GOSUB 2020
1220 NEXT X,Y
1230 HCOLOR= 3
1240 REM % & 0
1250 FOR Y = 33 TO 133 STEP 20
1260 X = 31: GOSUB 2080
```

```
1270 X = 37: GOSUB 2380
1280 NEXT Y
1290 REM 2 TO 8
1300 X = 25: Y = 113
1310 FOR I = 1 TO 4
1320 ON I GOSUB 2150, 2200, 2240, 2290
1330 Y = Y - 20
1340 NEXT I
1350 GOSUB 2080
1360 REM 1
1370 X = 20: Y = 33: GOSUB 2120
1380 RETURN
1390 REM X-AXIS
1400 FOR I = 1 TO K
1410 X = 47 + 20 * I: Y = 143: GOSUB 2020
1420 HCOLOR= 3: X = X + 1: Y = Y - 1
1430 ON I GOSUB 2450, 2490, 2540, 2580, 2620, 2
    660, 2690, 2740, 2780, 2820
1440 NEXT I
1450 RETURN
1460 REM BARS
1470 REM COMPUTE HEIGHT
1480 GOSUB 1520
1490 REM PLOT
1500 GOSUB 1620
1510 RETURN
1520 REM HEIGHT
1530 REM PERCENTAGES
1540 FOR I = 1 TO K
1550 PCT(I) = CLASS(I) * 100 / SUM
1560 NEXT
1570 REM HEIGHT
1580 FOR I = 1 TO K
1590 Y(I) = 130 - PCT(I)
1600 NEXT
1610 RETURN
1620 REM PLOT
1630 HCOLOR= C2: X = 65
1640 FOR I = 1 TO K
1650 FOR J = X TO X + 10
1660 HPLLOT J, 130 TO J, Y(I)
1670 NEXT J
1680 X = X + 20
1690 NEXT I
1700 RETURN
1710 REM TITLE
1720 L = LEN (TITLE$)
1730 VTAB 21: HTAB (21 - L / 2): PRINT TIT
    LE$
1740 RETURN
```

```
1750 REM ANALYZE
1760 VTAB 23: HTAB 1: PRINT SPC( 40): VTAB
24: HTAB 14: PRINT SPC( 14)
1770 VTAB 23: HTAB 1: PRINT "Analyze a Bar
(Y/N) ? "; CHR$( 7);
1780 GET S$
1790 IF S$ = "N" OR S$ = "n" THEN 1880
1800 IF S$ < > "Y" AND S$ < > "y" THEN 1
770
1810 REM CHOOSE
1820 GOSUB 1890
1830 REM DISPLAY RESULT
1840 GOSUB 1980
1850 VTAB 24: HTAB 14: INVERSE : PRINT "PR
ESS ANY KEY ";: NORMAL
1860 GET S$
1870 GOTO 1760
1880 RETURN
1890 REM CHOOSE
1900 VTAB 23: HTAB 1: PRINT SPC( 22)
1910 VTAB 23: HTAB 1: PRINT "Which one ? "
; CHR$( 7);
1920 GET S$
1930 A = ASC (S$)
1940 IF A > 90 THEN A = A - 32
1950 N = A - 64
1960 IF N < 1 OR N > K THEN 1910
1970 RETURN
1980 REM RESULT
1990 VTAB 23: HTAB 1: PRINT SPC( 12)
2000 VTAB 23: HTAB 1: PRINT CHR$( A);": C
ount= ";CLASS(N);" ("; INT (PCT(N));"%
of total)"
2010 RETURN
2020 REM BLACK BACKGROUND FOR LETTER
2030 HCOLOR= 0
2040 FOR L = X TO X + 6
2050 HPLLOT L,Y TO L,Y - 8
2060 NEXT L
2070 RETURN
2080 REM 0
2090 HPLLOT X,Y - 1 TO X,Y - 5: HPLLOT X + 4
,Y - 1 TO X + 4,Y - 5
2100 HPLLOT X + 1,Y TO X + 3,Y: HPLLOT X + 1
,Y - 6 TO X + 3,Y - 6
2110 RETURN
2120 REM 1
2130 HPLLOT X + 1,Y TO X + 3,Y: HPLLOT X + 2
,Y TO X + 2,Y - 6: HPLLOT X + 1,Y - 5
2140 RETURN
```

```
2150 REM 2
2160 HPLOT X,Y TO X + 4,Y: HPLLOT X,Y - 1: HPLLOT
X + 1,Y - 2
2170 HPLLOT X + 2,Y - 3 TO X + 3,Y - 3: HPLLOT
X + 4,Y - 4 TO X + 4,Y - 5
2180 HPLLOT X + 1,Y - 6 TO X + 3,Y - 6: HPLLOT
X,Y - 5
2190 RETURN
2200 REM 4
2210 HPLLOT X + 3,Y TO X + 3,Y - 6: HPLLOT X
,Y - 2 TO X + 4,Y - 2
2220 HPLLOT X,Y - 3: HPLLOT X + 1,Y - 4: HPLLOT
X + 2,Y - 5
2230 RETURN
2240 REM 6
2250 HPLLOT X + 1,Y TO X + 3,Y: HPLLOT X,Y -
3 TO X + 3,Y - 3: HPLLOT X + 2,Y - 6 TO
X + 4,Y - 6
2260 HPLLOT X + 1,Y - 5: HPLLOT X,Y - 4
2270 HPLLOT X,Y - 1 TO X,Y - 2: HPLLOT X + 4
,Y - 1 TO X + 4,Y - 2
2280 RETURN
2290 REM 8
2300 FOR J = 0 TO 6 STEP 3
2310 HPLLOT X + 1,Y - J TO X + 3,Y - J
2320 NEXT J
2330 FOR J = 0 TO 4 STEP 4
2340 HPLLOT X + J,Y - 1 TO X + J,Y - 2
2350 HPLLOT X + J,Y - 4 TO X + J,Y - 5
2360 NEXT J
2370 RETURN
2380 REM %
2390 FOR J = 1 TO 5: HPLLOT X + J - 1,Y - J
: NEXT J
2400 FOR J = 0 TO 1
2410 HPLLOT X + J,Y - 5 TO X + J,Y - 6
2420 HPLLOT X + 3 + J,Y TO X + 3 + J,Y - 1
2430 NEXT J
2440 RETURN
2450 REM A
2460 HPLLOT X,Y TO X,Y - 4: HPLLOT X + 4,Y TO
X + 4,Y - 4
2470 HPLLOT X,Y - 2 TO X + 4,Y - 2: HPLLOT X
+ 1,Y - 5: HPLLOT X + 3,Y - 5: HPLLOT X
+ 2,Y - 6
2480 RETURN
2490 REM B
2500 HPLLOT X,Y TO X,Y - 6: HPLLOT X,Y - 6 TO
X + 3,Y - 6
```



```
2510 HPLOT X,Y TO X + 3,Y: HPLOT X,Y - 3 TO
X + 3,Y - 3
2520 HPLOT X + 4,Y - 5: HPLOT X + 4,Y - 4:
HPLOT X + 4,Y - 1: HPLOT X + 4,Y - 2
2530 RETURN
2540 REM C
2550 HPLOT X,Y - 1 TO X,Y - 5: HPLOT X + 1
,Y - 6 TO X + 3,Y - 6
2560 HPLOT X + 1,Y TO X + 3,Y: HPLOT X + 4
,Y - 5: HPLOT X + 4,Y - 1
2570 RETURN
2580 REM D
2590 HPLOT X,Y TO X,Y - 6: HPLOT X,Y TO X +
3,Y
2600 HPLOT X,Y - 6 TO X + 3,Y - 6: HPLOT X
+ 4,Y - 1 TO X + 4,Y - 5
2610 RETURN
2620 REM E
2630 HPLOT X + 4,Y TO X,Y: HPLOT TO X,Y -
6
2640 HPLOT TO X + 4,Y - 6: HPLOT X,Y - 3 TO
X + 3,Y - 3
2650 RETURN
2660 REM F
2670 HPLOT X,Y TO X,Y - 6: HPLOT TO X + 4
,Y - 6: HPLOT X,Y - 3 TO X + 3,Y - 3
2680 RETURN
2690 REM G
2700 HPLOT X,Y - 1 TO X,Y - 5: HPLOT X + 1
,Y - 6 TO X + 4,Y - 6
2710 HPLOT X + 1,Y TO X + 4,Y: HPLOT X + 4
,Y - 1
2720 HPLOT X + 4,Y - 2: HPLOT X + 3,Y - 2
2730 RETURN
2740 REM H
2750 HPLOT X,Y TO X,Y - 6: HPLOT X + 4,Y TO
X + 4,Y - 6
2760 HPLOT X,Y - 3 TO X + 4,Y - 3
2770 RETURN
2780 REM I
2790 HPLOT X + 2,Y TO X + 2,Y - 6
2800 HPLOT X + 1,Y TO X + 3,Y: HPLOT X + 1
,Y - 6 TO X + 3,Y - 6
2810 RETURN
2820 REM J
2830 HPLOT X + 4,Y - 1 TO X + 4,Y - 6
2840 HPLOT X,Y - 1
2850 HPLOT X + 1,Y TO X + 3,Y
2860 RETURN
```

Analysis of Variance

Analyzing variance, the dispersion of observations about a mean, can show if a connection between the observations actually exists.

So you want to use your hard-earned dollars on Wall Street. But how do you know which stock will continue to do well? Using statistical inquiry, you can find out. It's better than tossing darts at a board.

"Analysis of Variance" can help you determine if there is a connection between variances or if they're only random.

Trends

Enter and save the program, then type RUN. As an example, we'll look at three groups of stocks with five stocks in each group.

Refer to the table. The average percentage changes in the price of 15 utility, airline, and computer stocks are listed.

Changes in Returns

Average Annual Change in Stock Prices			
	Utilities	Airline Stocks	Computer Stocks
	3.7%	2.2%	12.8%
	3.0	3.1	9.7
	4.3	2.8	10.3
	5.1	1.9	13.0
	<u>4.4</u>	<u>3.0</u>	<u>10.2</u>
Sample Mean	4.1%	2.6%	11.2%

Our job is to decide if the differences in average returns in each industry are due to chance or to some systematic cause, like rate of growth in sales. If it's the latter, we'd want to invest. If it's just luck that has increased the average returns, we probably wouldn't want to buy.

The Analysis of Variance program clears the fog from this problem. When you use the routine, first enter the number of samples (3 in the example, representing the three groups of stock), followed by the number of observations in each (5, to

represent the five stocks in each group). The default limits of ten samples and 30 observations in each sample can be changed by altering line 230 in the program. Now enter the data from the table, sample by sample.

The computer spends a moment calculating and then displays something like the figure below on the screen. While perhaps a bit imposing at first, this analysis of variance is actually pretty straightforward.

Analysis of Variance

Analysis of Variance		
Source of Variation	Sum of Squares	Degrees of Freedom
Among sample	211.033	2
Within sample	13.460	12
Total	224.493	14
F-Statistic = 94.071		

Note that the average return on the computer stocks is a lot higher than the average return on utilities or airlines. Also, notice that the fluctuations of percentages *within* each separate sample is relatively small. The *within sample* sum of squares measures this deviation. Since the within sample figure is so different from the *among sample* value, one can assume that in general (using these hypothetical figures), the three types of stock differ significantly in performance.

The *degrees of freedom* figure is the difference between the number of observations and the number of calculations. In the among sample category, the three samples are compared to the grand mean, so there are 2 degrees of freedom (3 observations minus 1 calculation). There are 3 calculations in the within sample total (one for each of the three samples) and 15 observations, so there are 12 degrees of freedom ($15 - 3 = 12$).

F-statistic refers to the result of dividing the among sample sum of squares (adjusted by the degrees of freedom) by the within sample sum of squares (also adjusted by the degrees of freedom). The greater the F-statistic, the more variance between samples. If you're looking for significant differences between samples, you would hope to see a large number here.

Additional Analysis

You can use this program to analyze other things besides stock returns, of course. Any group of numbers, whether precipitation amounts for different months or sales per week, could be looked at with this program. As long as the numbers are separated into samples, or groups, you can quickly find out if there is any connection between the figures, or if a pattern is just coincidental.

Analysis of Variance

```
100 REM ANALYSIS OF VARIANCE
110 REM INITIALIZE
120 GOSUB 200
130 REM ENTER DATA
140 GOSUB 690
150 REM COMPUTE
160 GOSUB 800
170 REM DISPLAY RESULTS
180 GOSUB 1270
190 END
200 REM INITIALIZE
210 PRINT CHR$(21): TEXT : HOME
220 REM MAXIMUM NUMBERS OF SAMPLES & OBSERVATIONS (per sample)
230 DATA 10,30
240 READ KX,NX
250 DIM X(NX,KX)
260 REM GREETING
270 GOSUB 330
280 REM NUMBER OF SAMPLES
290 GOSUB 500
300 REM SAMPLE SIZE
310 GOSUB 590
320 RETURN
330 REM GREETING
340 HOME
350 PRINT " This program performs an Analysis of
360 PRINT "Variance on your sample data.
370 PRINT
380 PRINT " The maximum numbers of samples and
390 PRINT "observations (per sample) allowed are:
400 PRINT
410 PRINT TAB(17)"Samples = ";KX
420 PRINT TAB(12)"Observations = ";NX
```

```
430 PRINT : PRINT " Change line 230 for d
    ifferent values.
440 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT I
450 REM DIGITS AFTER DECIMAL
460 DA = 3
470 VTAB 23: HTAB 14: PRINT "Press any key
    ";
480 GET S$
490 RETURN
500 REM NUMBER OF SAMPLES
510 HOME
520 VTAB 1: HTAB 32: PRINT SPC( 8); CHR$
    (7)
530 VTAB 1: HTAB 1: INPUT "How many sample
    s do you have ? ";K$
540 K = VAL (K$)
550 VTAB 23: HTAB 8: PRINT SPC( 32)
560 IF K < 2 THEN VTAB 23: HTAB 8: PRINT
    "At least two are needed !": GOTO 520
570 IF K > KX THEN VTAB 23: HTAB 10: PRINT
    "Only ";KX;" are allowed !": GOTO 520
580 RETURN
590 REM SAMPLE SIZE
600 HOME
610 PRINT "How many observations per sampl
    e do you
620 VTAB 2: HTAB 8: PRINT SPC( 10); CHR$
    (7)
630 VTAB 2: HTAB 1: INPUT "have ? ";N$
640 N = VAL (N$)
650 VTAB 23: HTAB 8: PRINT SPC( 32)
660 IF N < 2 THEN VTAB 23: HTAB 8: PRINT
    "At least two are needed !": GOTO 620
670 IF N > NX THEN VTAB 23: HTAB 10: PRINT
    "Only ";NX;" are allowed !": GOTO 620
680 RETURN
690 REM ENTER DATA
700 FOR I = 1 TO K
710 HOME
720 PRINT "Please enter data on Sample ";I
    ;"."; CHR$ (7)
730 FOR J = 1 TO N
740 VTAB 3: HTAB 11: PRINT SPC( 20)
750 VTAB 3: HTAB 1: PRINT "No. ";J; TAB( 8
    );"=" ";
760 INPUT V$
770 X(J,I) = VAL (V$)
780 NEXT J,I
790 RETURN
```

```
800 REM CALCULATIONS
810 HOME
820 VTAB 12: HTAB 15: FLASH : PRINT "COMPU
      TING";: NORMAL
830 REM PRELIMINARY
840 GOSUB 880
850 REM FINAL
860 GOSUB 1070
870 RETURN
880 REM PRELIMINARY
890 REM SAMPLE TOTALS
900 FOR I = 1 TO K
910 T(I) = 0
920 FOR J = 1 TO N
930 T(I) = T(I) + X(J,I)
940 NEXT J,I
950 REM GRAND TOTAL
960 GD = 0
970 FOR I = 1 TO K
980 GD = GD + T(I)
990 NEXT
1000 REM SUM OF OBSERVATIONS SQUARED
1010 SQ = 0
1020 FOR I = 1 TO K
1030 FOR J = 1 TO N
1040 SQ = SQ + X(J,I) * X(J,I)
1050 NEXT J,I
1060 RETURN
1070 REM FINAL
1080 REM KEY TERMS
1090 T1 = GD * GD / (K * N)
1100 T2 = 0
1110 FOR I = 1 TO K
1120 T2 = T2 + T(I) * T(I)
1130 NEXT
1140 T2 = T2 / N
1150 T3 = SQ
1160 REM SUMS OF SQUARES
1170 TL = T3 - T1
1180 AMG = T2 - T1
1190 WITHN = T3 - T2
1200 REM F-STATISTIC
1210 REM NUMERATOR
1220 F1 = AMG / (K - 1)
1230 REM DENOMINATOR
1240 F2 = WITHN / (K * N - K)
1250 F = F1 / F2
1260 RETURN
1270 REM DISPLAY RESULTS
1280 HOME
```

```

1290 PRINT L$
1300 PRINT TAB(10)"ANALYSIS OF VARIANCE
1310 PRINT L$
1320 VTAB 4: HTAB 33: INVERSE : PRINT "DEG
REES
1330 VTAB 5: HTAB 1: PRINT "SOURCE OF";: HTAB
33: PRINT " OF "
1340 VTAB 6: HTAB 1: PRINT "VARIATION";: HTAB
16: PRINT "SUM OF SQUARES";: HTAB 33: PRINT
"FREEDOM": NORMAL
1350 REM VALUES
1360 GOSUB 1410
1370 VTAB 22: HTAB 1: PRINT L$
1380 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
1390 GET S$
1400 RETURN
1410 REM VALUES
1420 VTAB 8: HTAB 1: PRINT "Among Sample";
:HT = 29:NR = AMG: GOSUB 9000
1430 VTAB 8:HT = 37:DA = 0:NR = K - 1: GOSUB
9000:DA = 3
1440 VTAB 10: HTAB 1: PRINT "Within Sample
";:HT = 29:NR = WITHN: GOSUB 9000
1450 VTAB 10:HT = 37:DA = 0:NR = K * N - K
: GOSUB 9000:DA = 3
1460 VTAB 12: HTAB 1: PRINT "Total";:HT =
29:NR = TL: GOSUB 9000
1470 VTAB 12:HT = 37:DA = 0:NR = K * N - 1
: GOSUB 9000:DA = 3
1480 VTAB 18: HTAB 1: PRINT "F-Statistic =
";:HT = 29:NR = F: GOSUB 9000
1490 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

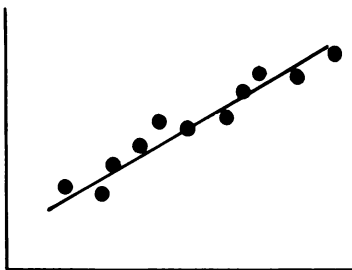
```

Simple Correlation Coefficients

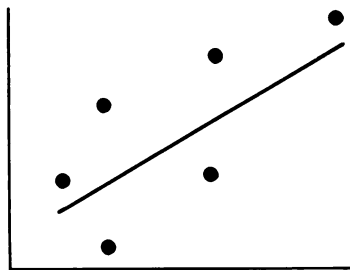
Measuring the association between two variables can show how strongly the variables are connected. The computation is simple with this program.

A simple correlation coefficient measures the degree of linear association between two variables. These coefficients are always between -1 and $+1$, inclusive. A value close to either extreme means that the linear relationship between the two terms is strong, while a correlation near 0 means that it's weak. Graphed, these kinds of correlations would look like the illustrations in Figure 1.

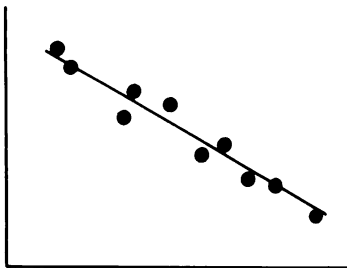
Figure 1. Linear Correlations



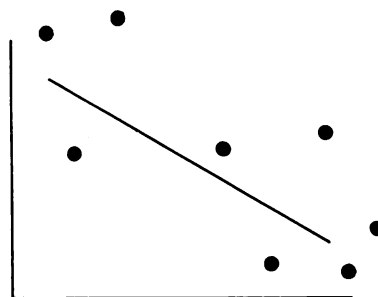
Strongly Positive



Weakly Positive



Strongly Negative



Weakly Negative

When the correlations are grouped tightly around the $+1$ or -1 line, as in the left-hand illustrations, there is a strong

relationship between the variables. If they're scattered far from the line, as in the right-hand illustrations, it shows a weaker relationship. You can put this idea to work for you anytime you want to establish the strength of the relationship between sets of variables. Here is an example.

Gold and Silver

Most of us have probably viewed the ups and downs of gold and silver prices in the past with a why-didn't-I-buy-then philosophy. The prices of each metal are listed in the table below.

Prices per Troy Ounce

**Prices Per Troy Ounce of Gold and Silver
in the United States**

Year	Price of Gold	Price of Silver
1967	\$ 35.0	\$ 1.5
1968	39.0	2.1
1969	41.5	1.8
1970	36.2	1.8
1971	41.0	1.6
1972	58.1	1.7
1973	96.5	2.6
1974	158.1	4.8
1975	161.7	4.4
1976	125.9	4.4
1977	147.4	4.6
1978	192.9	5.4
1979	303.6	10.7
1980	618.0	21.6
1981	458.7	10.7
1982	378.5	8.0
1983	407.9	11.6
1984	371.3	8.0

Source: From prices and indices compiled by the Bureau of Labor.

Regrets aside, a quick glance at the table suggests that the two sets of prices are strongly related. Both rose sharply in 1973 and 1974, and again in 1979 and 1980. Gold achieved its highest price at the same time silver did. The time periods for

lowest price also coincide. But just how strong is this relationship?

To find out, all you need to do is type in, save, and run "Simple Correlation Coefficients." You'll first be asked to enter the number of variables. In our example, there are only two variables—gold prices and silver prices. (If you want to change the default limits of the number of variables or the number of observations, alter program line 320.)

Then you simply enter the data on gold and silver prices from 1967 through 1984, or for 18 years (observations) in all. Use the figures you see in the table. You're given a chance to review and edit what you've input. The computer then displays the figures shown in Figure 2.

Figure 2. Simple Correlation Coefficients

	X1	X2
X1	1.000	0.952
X2	0.952	1.000

X1 represents the price of gold and X2 the price of silver. The simple linear correlation between gold and silver prices is approximately 0.95, a value very close to +1. Obviously, there is a strong relationship between the price of gold and the price of silver.

There are dangers to this kind of conclusion, however. A strong relationship between two variables, either positive or negative, can be a temptation to call one the cause and the other the effect. At times this is reasonable, but at other times it is ridiculous. Just because the sun rises after the cock crows doesn't mean that the crow causes the rising. A third variable may cause both. You should always rely upon common sense or well-established theory in determining which correlations are reasonable and which are coincidental.

An additional word of warning: only *linear* association is measured by the correlation coefficient. In a circle, for example, the linear relationship between X and Y is zero, while the circular association is perfect.

Notes on the Program

If the observations on a variable are all the same, then any simple correlation coefficient involving that variable cannot be computed since division by zero would be required. The computer program handles this as shown in Figure 3.

Figure 3. Same Observations on Variables

Data		Correlation Coefficients	
<u>X1</u>	<u>X2</u>	X1	X2
3	7	X1	undefined
3	8	X2	undefined
3	20		1

Simple Correlation Coefficients

```
100 REM CORRELATION COEFFICIENTS
110 REM INITIALIZE
120 GOSUB 220
130 REM ENTER DATA
140 GOSUB 630
150 REM EDIT DATA
160 GOSUB 950
170 REM COMPUTE COEFFICIENTS
180 GOSUB 1360
190 REM DISPLAY RESULTS
200 GOSUB 1590
210 END
220 REM INITIALIZE
230 REM MAXIMUM NUMBERS OF OBSERVATIONS &
    VARIABLES
240 GOSUB 300
250 REM ENTER NUMBER OF VARIABLES
260 GOSUB 460
270 REM CREATE SYMBOLS
280 GOSUB 550
290 RETURN
300 REM MAXIMUM VALUES
310 PRINT CHR$(21): TEXT : HOME
320 DATA 25,25: REM OBSERVATIONS, VARIABLE
    S
330 READ NX,KX
340 DIM V$(KX),R(KX,NX),S(KX),SS(KX),X(NX,
    KX)
350 PRINT "This program computes simple co
    rrelation";
360 PRINT "coefficients.
370 PRINT
380 PRINT "The maximum numbers of observat
    ions and
390 PRINT "variables allowed are
400 VTAB 7: HTAB 9: PRINT "Observations =
    ";NX
410 VTAB 8: HTAB 12: PRINT "Variables = ";
    KX
```

```

420  VTAB 10: HTAB 1: PRINT "Change line 32
    0 for different limits.
430  VTAB 23: HTAB 14: PRINT "Press any key
    ";
440  GET S$
450  RETURN
460  REM ENTER NO. OF VARIABLES
470  HOME
480  VTAB 1: HTAB 34: PRINT SPC( 6); CHR$
    (7)
490  VTAB 1: HTAB 1: INPUT "How many variab
    les do you have ? ";K$
500  K = INT ( VAL (K$))
510  VTAB 23: HTAB 1: PRINT SPC( 39);
520  IF K < 2 THEN VTAB 23: HTAB 8: PRINT
    "At least two are needed !": GOTO 480
530  IF K > KX THEN VTAB 23: HTAB 10: PRINT
    "Only ";KX;" are allowed !": GOTO 480
540  RETURN
550  REM CREATE SYMBOLS
560  FOR I = 1 TO K
570  V$(I) = "X" + STR$ (I)
580  NEXT
590  L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT
600  REM DIGITS AFTER DECIMAL
610  DA = 3
620  RETURN
630  REM ENTER DATA
640  REM ON THE FIRST VARIABLE
650  GOSUB 690
660  REM ON THE OTHERS
670  GOSUB 840
680  RETURN
690  REM FIRST VARIABLE
700  HOME
710  PRINT "Please enter observations for X
    1. Hit
720  PRINT "the RETURN key when you're thro
    ugh."; CHR$ (7)
730  N = NX
740  FOR J = 1 TO NX
750  VTAB 4: HTAB 12: PRINT SPC( 20)
760  VTAB 4: HTAB 1: PRINT "X1( ";J; TAB( 8
    );")= ";
770  INPUT X$
780  IF X$ = "" THEN N = J - 1:J = NX
790  IF X$ < > "" THEN X(J,1) = VAL (X$)
800  NEXT J
810  REM CHECK FOR TOO FEW OBSERVATIONS

```

```
820 IF N < 2 THEN VTAB 22: HTAB 2: PRINT
    "You need at least two observations !"
    ; CHR$ (7);: GOTO 730
830 RETURN
840 REM OTHER VARIABLES
850 FOR I = 2 TO K
860 HOME
870 PRINT "Please enter observations for "
    ;V$(I);". "; CHR$ (7)
880 FOR J = 1 TO N
890 VTAB 3: HTAB 13: PRINT SPC( 20)
900 VTAB 3: HTAB 1: PRINT V$(I); TAB( 4)"(
    ";J; TAB( 9)")= ";
910 INPUT X$
920 X(J,I) = VAL (X$)
930 NEXT J,I
940 RETURN
950 REM CORRECT DATA
960 FOR I = 1 TO K
970 FOR J = 1 TO N STEP 10
980 REM DISPLAY
990 GOSUB 1040
1000 REM CORRECT
1010 GOSUB 1170
1020 NEXT J,I
1030 RETURN
1040 REM DISPLAY DATA
1050 HOME
1060 PRINT L$
1070 A$ = CHR$ (32)
1080 IF J > 10 THEN A$ = " more "
1090 PRINT "These are";A$;"values of ";V$(
    I);"."
1100 PRINT L$: PRINT
1110 R = 0
1120 FOR L = J TO J + 9
1130 IF L < = N THEN R = R + 1: INVERSE :
    PRINT CHR$ (R + 64);: NORMAL : PRINT
    " No. ";L; TAB( 10)"= ";X(L,I)
1140 NEXT L
1150 VTAB 16: HTAB 1: PRINT L$
1160 RETURN
1170 REM CORRECT DATA
1180 VTAB 21: HTAB 1: PRINT SPC( 39)
1190 VTAB 23: HTAB 1: PRINT SPC( 39)
1200 VTAB 19: HTAB 1: PRINT "Corrections (
    Y/N) ? "; CHR$ (7);
1210 GET S$
1220 IF S$ = "N" OR S$ = "n" THEN 1350
```

```
1230 IF S$ < > "Y" AND S$ < > "y" THEN 1
200
1240 VTAB 21: HTAB 1: PRINT "Which letter
? "; CHR$ (7);
1250 GET S$
1260 A = ASC (S$)
1270 IF A > 90 THEN A = A - 32
1280 Q = A - 64
1290 IF Q < 1 OR Q > R THEN 1240
1300 VTAB (A - 60): HTAB 1: FLASH : PRINT
CHR$ (A); CHR$ (7);: NORMAL
1310 VTAB 23: HTAB 1: INPUT "What should t
he value be ? ";S$
1320 X(J + Q - 1,I) = VAL (S$)
1330 VTAB (A - 60): HTAB 12: PRINT SPC( 2
0);: HTAB 12: PRINT X(J + Q - 1,I)
1340 VTAB (A - 60): HTAB 1: INVERSE : PRINT
CHR$ (A);: NORMAL : GOTO 1180
1350 RETURN
1360 REM COMPUTE
1370 HOME
1380 VTAB 12: HTAB 15: FLASH : PRINT "COMP
UTING";: NORMAL
1390 REM SUM OF OBSERVATIONS & OBSERVATIO
NS SQUARED
1400 FOR I = 1 TO K
1410 S(I) = 0:SS(I) = 0
1420 FOR J = 1 TO N
1430 S(I) = S(I) + X(J,I)
1440 SS(I) = SS(I) + X(J,I) ^ 2
1450 NEXT J,I
1460 REM CORRELATION MATRIX
1470 FOR I = 1 TO K
1480 FOR J = I TO K
1490 SC = 0
1500 FOR L = 1 TO N
1510 SC = SC + X(L,I) * X(L,J)
1520 NEXT L
1530 V = (N * SS(I) - S(I) ^ 2) * (N * SS(J)
) - S(J) ^ 2)
1540 IF V < = 0 THEN R(I,J) = - 999
1550 IF V > 0 THEN R(I,J) = (N * SC - S(I)
* S(J)) / SQR (V):R(I,I) = 1
1560 R(J,I) = R(I,J)
1570 NEXT J,I
1580 RETURN
1590 REM DISPLAY RESULTS
1600 FOR Q = 1 TO K STEP 10
1610 FOR I = 1 TO K STEP 3
1620 REM HEADING
```

```

1630 GOSUB 1680
1640 REM BODY
1650 GOSUB 1800
1660 NEXT I,Q
1670 RETURN
1680 REM HEADING
1690 HOME
1700 PRINT L$
1710 PRINT TAB( 8)"SIMPLE CORRELATION MAT
    RIX
1720 PRINT L$
1730 REM COLUMN HEADING
1740 COL = 12
1750 FOR L = I TO I + 2
1760 IF L < = K THEN VTAB 5: HTAB COL: PRINT
    V$(L);
1770 COL = COL + 12
1780 NEXT L
1790 RETURN
1800 REM VALUES
1810 ROW = 8
1820 FOR J = Q TO Q + 9
1830 IF J > K THEN 1930
1840 VTAB ROW: HTAB 1: PRINT V$(J);
1850 COL = 14
1860 FOR L = I TO I + 2
1870 IF L > K THEN 1910
1880 IF R(J,L) < > - 999 THEN VTAB ROW:
    HT = COL:NR = R(J,L): GOSUB 9000
1890 IF R(J,L) = - 999 THEN VTAB ROW: HTAB
    COL - 8: PRINT "Undefined"
1900 COL = COL + 12
1910 NEXT L
1920 ROW = ROW + 1
1930 NEXT J
1940 VTAB 22: HTAB 1: PRINT L$
1950 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
1960 GET S$
1970 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)

```

```
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
    Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN
```


Simple Linear Regression Analysis

Regression analysis, the estimation of inverse relationships, can be just as effective in showing connections between variables as correlation coefficients. Some problems in simple linear regression analysis can be handled by the following program.

Estimating relationships between variables can take on many forms. Sometimes you want to establish that two things have something in common. Other times you'll want to show a different association. There still may be a connection, but it's based on another kind of relationship. One variable might increase every time another decreases. We call that an inverse relationship.

For example, suppose you want to estimate the relationship between stock prices and interest rates. Your guess may be that stock prices rise when interest rates fall, and that stock prices fall when interest rates rise. With "Simple Linear Regression Analysis," you can find out if this assertion is valid.

Ups and Downs

You first need to enter and save this program, and then run it. But you're still lacking some data to input. The table below, "Stock Prices and Interest Rates," provides it.

When you run the program, notice that you can alter the number of possible observations by changing line 250 in the program. The computer asks for the observations under first Y, then X. Go ahead and enter the data you see in the table (or your own numbers). In our example, X is the interest rate and Y is the stock price. Hit the RETURN key when you're through keying in the numbers for Y, and then begin typing in the observations for X. After data entry, you have a chance to review and edit the observations. The computer takes a few seconds to calculate the problem and then displays the results as shown in Figure 1.

Stock Prices and Interest Rates

Year and Quarter	Standard and Poor's Index of 500 Leading Stocks	3-Month T-Bill Rate, %
82:1	114.2	12.8
82:2	114.2	12.4
82:3	113.8	9.3
82:4	136.7	7.9
83:1	147.7	8.1
83:2	162.7	8.4
83:3	165.5	9.1
83:4	165.7	8.8
84:1	160.4	9.2
84:2	155.8	9.8
84:3	160.5	10.3

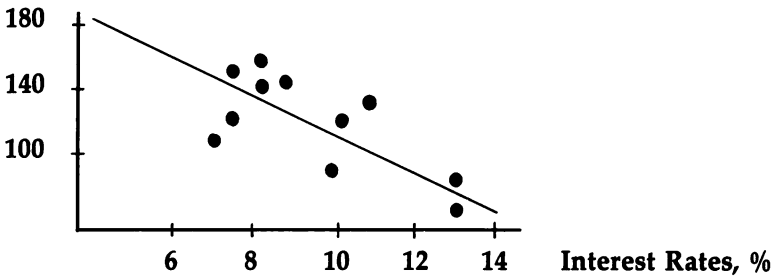
Figure 1. Regression Results

Term	Estimated Value	Standard Error
B0	221.070	35.237
B1	-7.867	3.607
Number of Observations	=	11.000
R-Squared	=	0.346
Standard Error of the Estimate	=	18.501
F-Statistic	=	4.756

Statistical Analysis

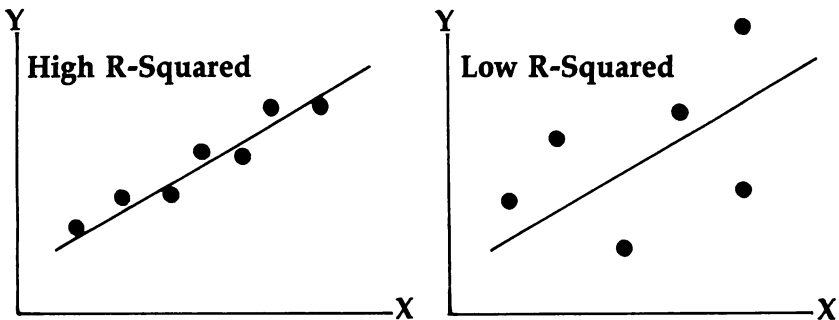
Two terms, a *constant* and a *coefficient*, are listed on the left as B0 and B1, respectively. The coefficient can also be considered the *slope*. The value of almost -7.9 for the slope means that if interest rates were to increase by 1 percentage point, stock prices would be expected to fall by almost 7.9 points. The inverse relationship between stock prices and interest rates holds, as suspected.

You can use the constant and coefficient to draw a trend line between the two variables, as Figure 2 shows.

Figure 2. Trend LineIndex of
Stock Prices

The *standard errors* of the constant and the coefficient measure the precision of each variable's estimated values. The smaller the number, the more precise the estimate.

The number of observations is listed, then three figures, which can be called *accuracy of fit* statistics. *R-squared*, or the coefficient of determination, is the proportion of variation in Y (stock prices) explained by X (interest rates). It ranges from 0 to 1, with a value close to 0 indicating a strong association. The higher the value, the better the regression line fits the data. Figure 3 illustrates this. Since the R-squared value is approximately 0.35, roughly a third of the change in stock prices is due to interest rates.

Figure 3. Goodness of Fit

The *standard error of the estimate*, about 18.5 in this instance, is the standard deviation of actual predicted values minus predicted Y values. It can be loosely interpreted as the

average error made in predicting Y using the regression equation. Finally, the *F-statistic* measures how well the regression equation explains the variation in Y .

Obviously, there is not room here to discuss the theory and application of regression analysis. Nevertheless, many regression problems do not require a fancy solution and can be handled with a routine like this.

Technical Notes

In simple linear regression analysis, the term to be explained or predicted is called the *dependent variable*. The term to do the explaining is called, naturally enough, the *explanatory*, or *independent variable*. *Simple* means that only one explanatory variable appears in the regression equation (two or more appear in multiple regression—a program for computing multiple regression follows this section). Further, the term *linear* indicates that this form of regression involves estimating a straight line through a set of observations on a dependent and independent variable.

The equation for a straight line is $Y = mX + b$. Usually, Y is plotted on the vertical axis, and X on the horizontal. The symbol b is the Y -intercept of the equation, and m is its slope, or the change in Y divided by the change in X .

The equation for the simple linear regression model is similar to that of the straight line, namely:

$$Y = a + bX + \text{random error}$$

where Y is the dependent variable and X is the explanatory variable. The values a and b are the Y -intercept and slope. The only difference between the regression equation and the straight line is the appearance of the random error term in the regression equation.

If the error term were absent from an equation, each pair of observations on Y and X would lie exactly on a straight line, and regression analysis would not be needed. An example of this type of nonrandom relationship is the formula for computing batting averages of baseball players:

$$\text{Batting Average} = \text{Number of Hits} / \text{Times at Bat}$$

But for other relationships, the association between Y and X is behavioral rather than exact. The random error term is present in the equation, and use of regression analysis is required.

The essence of the program is estimating a line which minimizes the sum of squared distances of observations from that line. This is equivalent to satisfying, with computer precision, the need that often arises to be able to draw a line through a plot of points in a manner that best reflects the apparent trend.

Simple Linear Regression Analysis

```
100 REM SIMPLE LINEAR REGRESSION
110 REM INITIALIZE
120 GOSUB 220
130 REM ENTER DATA
140 GOSUB 390
150 REM EDIT DATA
160 GOSUB 730
170 REM COMPUTE
180 GOSUB 1140
190 REM DISPLAY RESULTS
200 GOSUB 1490
210 END
220 REM INITIALIZE
230 PRINT CHR$(21): TEXT : HOME
240 REM MAXIMUM NUMBER OF OBSERVATIONS
250 DATA 101
260 READ NX
270 DIM X(NX,2)
280 V$(1) = "Y":V$(2) = "X"
290 PRINT TAB(8)"SIMPLE LINEAR REGRESSIO
N
300 PRINT : PRINT "A total of ";NX;" obser
vations on your
310 PRINT "equation are allowed.
320 VTAB 7: HTAB 1: PRINT "Change line 250
for a different limit.
330 VTAB 23: HTAB 14: PRINT "Press any key
";
340 GET S$
350 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
NEXT
360 REM DIGITS AFTER DECIMAL
370 DA = 3
380 RETURN
390 REM ENTER DATA
400 REM ON Y
410 GOSUB 450
420 REM ON X
430 GOSUB 620
440 RETURN
```

```
450 REM Y
460 HOME
470 PRINT "Please enter observations on the
480 PRINT "dependent variable, Y. Hit the
RETURN
490 PRINT "key when through."; CHR$(7)
500 N = NX
510 FOR J = 1 TO NX
520 VTAB 5: HTAB 11: PRINT SPC(20)
530 VTAB 5: HTAB 1: PRINT "Y( ";J; TAB(7)
";")= ";
540 INPUT V$
550 IF V$ = "" THEN N = J - 1:J = NX
560 IF V$ < > "" THEN X(J,1) = VAL(V$)
570 NEXT J
580 REM DEGREES OF FREEDOM
590 V = N - 2
600 IF V < 1 THEN VTAB 22: HTAB 1: PRINT
"You need at least three observations
!"; CHR$(7);: GOTO 500
610 RETURN
620 REM X
630 HOME
640 PRINT "Please enter data on the explanatory
650 PRINT "variable, X."; CHR$(7)
660 FOR J = 1 TO N
670 VTAB 4: HTAB 11: PRINT SPC(20)
680 VTAB 4: HTAB 1: PRINT "X( ";J; TAB(7)
";")= ";
690 INPUT X$
700 X(J,2) = VAL(X$)
710 NEXT J
720 RETURN
730 REM CORRECT DATA
740 FOR I = 1 TO 2
750 FOR J = 1 TO N STEP 10
760 REM DISPLAY
770 GOSUB 820
780 REM CORRECT
790 GOSUB 950
800 NEXT J,I
810 RETURN
820 REM DISPLAY DATA
830 HOME
840 PRINT L$
850 A$ = CHR$(32)
860 IF J > 10 THEN A$ = " more "
```

```
870 PRINT "These are";A$;"values of ";V$(I
);";"
880 PRINT L$: PRINT
890 R = 0
900 FOR L = J TO J + 9
910 IF L < = N THEN R = R + 1: INVERSE : PRINT
CHR$(R + 64);: NORMAL : PRINT " No.
";L; TAB( 10)"= ";X(L,I)
920 NEXT L
930 VTAB 16: HTAB 1: PRINT L$
940 RETURN
950 REM CORRECT DATA
960 VTAB 21: HTAB 1: PRINT SPC( 39)
970 VTAB 23: HTAB 1: PRINT SPC( 39)
980 VTAB 19: HTAB 1: PRINT "Corrections (Y
/N) ? "; CHR$( 7);
990 GET S$
1000 IF S$ = "N" OR S$ = "n" THEN 1130
1010 IF S$ < > "Y" AND S$ < > "y" THEN 9
80
1020 VTAB 21: HTAB 1: PRINT "Which letter
? "; CHR$( 7);
1030 GET S$
1040 A = ASC (S$)
1050 IF A > 90 THEN A = A - 32
1060 Q = A - 64
1070 IF Q < 1 OR Q > R THEN 1020
1080 VTAB (A - 60): HTAB 1: FLASH : PRINT
CHR$( A); CHR$( 7);: NORMAL
1090 VTAB 23: HTAB 1: INPUT "What should t
he value be ? ";S$
1100 X(J + Q - 1,I) = VAL (S$)
1110 VTAB (A - 60): HTAB 12: PRINT SPC( 2
0);: HTAB 12: PRINT X(J + Q - 1,I)
1120 VTAB (A - 60): HTAB 1: INVERSE : PRINT
CHR$( A);: NORMAL : GOTO 960
1130 RETURN
1140 REM COMPUTE
1150 HOME
1160 VTAB 12: HTAB 15: FLASH : PRINT "COMP
UTING";: NORMAL
1170 REM KEY SUMS
1180 GOSUB 1240
1190 REM COEFFICIENT & CONSTANT
1200 GOSUB 1340
1210 REM ANOVA TERMS
1220 GOSUB 1380
1230 RETURN
1240 REM KEY SUMS
```

```

1250 SX = 0:SY = 0:XQ = 0:YQ = 0:CP = 0
1260 FOR I = 1 TO N
1270 SX = SX + X(I,2)
1280 SY = SY + X(I,1)
1290 XQ = XQ + X(I,2) ^ 2
1300 YQ = YQ + X(I,1) ^ 2
1310 CP = CP + X(I,1) * X(I,2)
1320 NEXT
1330 RETURN
1340 REM COEFFICIENT & CONSTANT
1350 B = (N * CP - SX * SY) / (N * XQ - SX *
    SX)
1360 A = (SY - B * SX) / N
1370 RETURN
1380 REM ANOVA TERMS
1390 TSS = YQ - SY * SY / N
1400 RSS = B * (CP - SX * SY / N)
1410 ESS = TSS - RSS
1420 REM ERROR VARIANCE & STANDARD ERROR
    OF THE ESTIMATE
1430 EV = ESS / V
1440 SEE = SQR (EV)
1450 REM STANDARD ERRORS OF THE COEFFICIE
    NT & CONSTANT
1460 SB = SQR (EV / (XQ - SX * SX / N))
1470 SA = SQR (EV * XQ / (N * XQ - SX * SX
    ))
1480 RETURN
1490 REM DISPLAY RESULTS
1500 REM EQUATION
1510 GOSUB 1550
1520 REM ANOVA TERMS
1530 GOSUB 1690
1540 RETURN
1550 REM EQUATION
1560 HOME
1570 PRINT L$
1580 PRINT TAB( 11)"REGRESSION RESULTS
1590 PRINT L$
1600 VTAB 5: HTAB 15: PRINT "Estimated"; TAB(
    32)"Standard
1610 VTAB 6: HTAB 1: PRINT "Term"; TAB( 17
    )"Value"; TAB( 34)"Error
1620 VTAB 8: HTAB 1: PRINT "B0"
1630 VTAB 8:HT = 23:NR = A: GOSUB 9000
1640 VTAB 8:HT = 39:NR = SA: GOSUB 9000
1650 VTAB 9: HTAB 1: PRINT "B1"
1660 VTAB 9:HT = 23:NR = B: GOSUB 9000
1670 VTAB 9:HT = 39:NR = SB: GOSUB 9000
1680 RETURN

```



```
1690 REM ANOVA TERMS
1700 VTAB 11: HTAB 1: PRINT "Number of": PRINT
    "Observations"; TAB( 17); "=";:HT = 32:
    NR = N: GOSUB 9000
1710 VTAB 14: HTAB 1: PRINT "R-Squared"; TAB(
    17) "=";:NR = RSS / TSS: GOSUB 9000
1720 VTAB 16: HTAB 1: PRINT "Standard Error
    r
1730 VTAB 17: HTAB 1: PRINT "of the Estima
    te"; TAB( 17) "=";:NR = SEE: GOSUB 9000

1740 VTAB 19: HTAB 1: PRINT "F-Statistic";
    TAB( 17) "=";:NR = RSS / EV: GOSUB 900
    0
1750 VTAB 21: HTAB 1: PRINT L$
1760 VTAB 23: HTAB 14: PRINT "Press any ke
    y ";
1770 GET S$
1780 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "": ZS$ = "": ZD$ = "": ZZ$ = "0000
    00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
    1)
9060 LET ZL = INT (ZN): ZR = INT ((ZN - Z
    L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
    Z$ + STR$ (ZR), DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3, 1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN
```

Multiple Linear Regression Analysis

"Multiple Linear Regression Analysis" estimates inverse relationships between up to six variables.

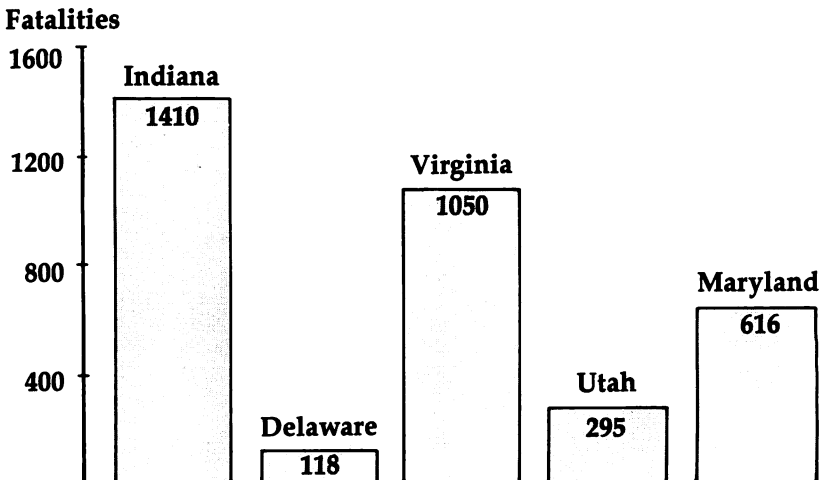
Using a statistical program to show the inverse relationship between variables, as you did in the program "Simple Linear Regression Analysis," is all well and good, as long as you have only two variables to consider. If you have a more complicated set of values, however, with several variables, you need something more powerful. This program can estimate the connection between up to six different variables, with up to 50 observations. It should be enough for almost any inverse trend you want to establish.

As always, an example is probably the best way to show how this program works. Let's go through one that's fairly complex.

Traffic Fatalities

The number of traffic fatalities varies from state to state. In 1964, for example, 1050 people died on highways in Virginia, while 118 died in Delaware, as Figure 1 shows. Why the difference? Is it simply that more people live in Virginia, or is there another reason?

Figure 1. 1964 Traffic Fatalities



You can use "Multiple Linear Regression Analysis" to try to find out. Your first job with this example, or with any problem of your own, is to think of variables that might explain the cause. In our example, we are investigating the possible root of a state's fatality count. We'll investigate two things: the number of automobile drivers in a state and the state's rural road mileage. You would expect fatalities to be highest in those states with more drivers and more rural roads.

Next, you need to gather observations on the variables. Using the example of traffic fatalities, you might come up with something like the table below.

Selected Data

State	Deaths in 1964	Drivers (10,000)	Rural Road Mileage (1000)
Delaware	118	30	3.4
Washington, D.C.	115	35	0
Indiana	1410	254	89
Iowa	833	150	100
Kansas	669	136	124
Maryland	616	157	29
Massachusetts	766	255	17
Mississippi	648	85	59
Nevada	215	23	44
New Mexico	387	54	62
South Dakota	270	40	87
Utah	295	57	32
Vermont	131	20	13
Virginia	1050	208	50
Washington	730	160	59

Source: D. F. Andrews, "Car Accidents—Environmental Aspects," *International Statistics Review*, vol. 41, 1973, as reprinted in N. R. Draper and H. Smith, *Applied Regression Analysis*, 2d ed. (New York: Wiley-Interscience, 1981), p. 191.

Using the Program

Assuming you've typed in, saved, and run the program, you can now enter the data into the computer. The computer first asks for the number of *explanatory variables*, or X's. Enter 2 for the traffic fatalities example; it uses two things to explain traffic fatalities (number of drivers and rural road mileage).

The program requests the observations on the *dependent* variable (specified as Y), or traffic fatalities. Key in 118 for Delaware, 115 for Washington, D.C., and so on, right down the list. After the figure for Washington State, press the RETURN key to signal the program that it's the last entry. You can then type in the observations on the X's in the same way.

After a few seconds of computing, the program displays:

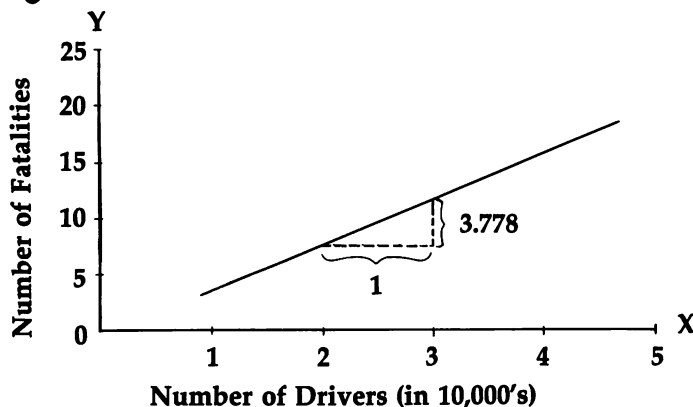
Figure 2. Regression Equation

Regression Equation			
Term	Estimated Value	Standard Error	T-Ratio
B0	-11.32879	66.71540	-0.16981
B1	3.77815	0.43170	8.75186
B2	2.77992	0.97588	2.84864

Press Any Key

B0 stands for the constant term in our regression equation. That's the number of fatalities in 1964. B1 and B2 correspond to the two X's (number of drivers, X1, and rural road mileage, X2, respectively). Both B1 and B2 measure the impact on Y of a one-unit change in the value of the corresponding X. Hence, the value of B1 means that 3.778 more people are expected to die in auto accidents over the course of a year for each additional 10,000 drivers in a state (X1 is measured in units of 10,000). Figure 3 shows the graph of this concept. The slope of the line, of the change in Y divided by the change in X, is 3.778.

Figure 3. Fatalities and the Number of Drivers



Similarly, the value of B2 indicates that 2.780 additional fatalities per year are expected if an additional 1000 miles of rural roads are constructed. With just a glance, you can see the effect of each explanatory variable on the number of traffic fatalities.

Other figures displayed on the screen are the *standard errors* and *t-ratios*. The lower the standard error, the more exact the estimation. The t-ratio is simply the estimated value divided by the standard error.

More Information

After digesting these sobering statistics, pressing any key shifts the display to the information shown in Figure 4.

Figure 4. Summary Values

Variation in Y	Sums of Squares	Degrees of Freedom
Total	2010494.408	14
Regression	1812250.140	2
Residual	198244.269	12
R-Squared	=	0.901
R-Bar Squared	=	0.885
F-Statistic	=	54.849
Standard Error of the Estimate	=	128.532

Some of these statistics are familiar from the simple linear regression example. New ones include the *total*, *regression*, and *residual sums of squares*. The total sums of squares measures the dispersion of observations on Y about the mean. The regression and residual sums of squares measure the dispersion, explained and unexplained, by the X's, respectively.

The *R-squared* value of 0.901 indicates that 90.1 percent of the variation in traffic fatalities from state to state is explained by the equation. *R-bar squared* (which is sometimes negative) is the R-squared statistic adjusted for degrees of freedom. It is useful in assessing the relative strengths of different models for explaining the same Y. As in the previous program, *F-statistic* measures how well the equation explains the variation in Y. Similarly, the *standard error of the estimate* represents the standard deviation of actual minus predicted Y values.

Two final statistics are displayed by the regression program:

Durbin-Watson statistic = 2.214
Rho = -0.121

Rho measures the degree of linear relationship between successive regression residuals. And the Durbin-Watson statistic is used to test for this kind of correlation.

In summary, multiple linear regression analysis is a straightforward extension of the simple linear case. In simple linear regression, *simple* means that only one explanatory variable appears in the equation, and *linear* means that a straight line is fitted between X and Y . In multiple regression, on the other hand, two or more explanatory variables are present, with *linear* now meaning that a plane (in the case of two X 's) is estimated instead of a line.

Multiple Linear Regression Analysis

```
100 REM MULTIPLE LINEAR REGRESSION
110 REM INITIALIZE
120 GOSUB 220
130 REM ENTER DATA
140 GOSUB 660
150 REM EDIT DATA
160 GOSUB 1010
170 REM COMPUTE
180 GOSUB 1420
190 REM DISPLAY RESULTS
200 GOSUB 3160
210 END
220 REM INITIALIZE
230 REM MAXIMUM NUMBERS OF OBSERVATIONS &
    VARIABLES
240 GOSUB 300
250 REM ENTER NUMBER OF VARIABLES
260 GOSUB 470
270 REM CREATE SYMBOLS
280 GOSUB 560
290 RETURN
300 REM MAXIMUM VALUES
310 PRINT CHR$(21): TEXT : HOME
320 DATA 50,6: REM OBSERVATIONS, EXPLANATORY VARIABLES
330 READ NX,KX
340 M = KX + 1
```

```
350 DIM B(M),C(M),B$(M),V$(M),R(M,M),V(M,M),Q(NX,M),X(NX,M)
360 PRINT "This program estimates a multiple linear";
370 PRINT "regression equation.
380 PRINT
390 PRINT "The maximum numbers of observations and
400 PRINT "explanatory variables (X's) allowed are:";
410 VTAB 7: HTAB 9: PRINT "Observations = ";NX
420 VTAB 8: HTAB 12: PRINT "Variables = ";KX
430 VTAB 10: HTAB 1: PRINT "Change line 32 0 for different limits.
440 VTAB 23: HTAB 14: PRINT "Press any key ";
450 GET S$
460 RETURN
470 REM ENTER NO. OF VARIABLES
480 HOME
490 VTAB 1: HTAB 28: PRINT SPC( 10); CHR$(7)
500 VTAB 1: HTAB 1: INPUT "How many X's do you have ? ";K$
510 K = INT ( VAL (K$))
520 VTAB 23: HTAB 1: PRINT SPC( 39);
530 IF K < 1 THEN VTAB 23: HTAB 8: PRINT "At least one is needed !": GOTO 490
540 IF K > KX THEN VTAB 23: HTAB 10: PRINT "Only ";KX;" are allowed !": GOTO 490
550 RETURN
560 REM CREATE SYMBOLS
570 V$(0) = "Y":B$(0) = "B0"
580 FOR I = 1 TO K
590 V$(I) = "X" + STR$(I)
600 B$(I) = "B" + STR$(I)
610 NEXT
620 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=": NEXT
630 REM DIGITS AFTER DECIMAL
640 DA = 3
650 RETURN
660 REM ENTER DATA
```

```

670 REM ON Y
680 GOSUB 720
690 REM ON THE X'S
700 GOSUB 900
710 RETURN
720 REM Y
730 HOME
740 PRINT "Please enter observations on th
e
750 PRINT "dependent variable, Y. Hit the
RETURN
760 PRINT "key when you're through."; CHR$
(7)
770 N = NX
780 FOR J = 1 TO NX
790 VTAB 5: HTAB 11: PRINT SPC( 20)
800 VTAB 5: HTAB 1: PRINT "Y( ";J; TAB( 7)
;")= ";
810 INPUT V$
820 IF V$ = "" THEN N = J - 1:J = NX
830 IF V$ < > "" THEN X(J,0) = VAL (V$)
840 NEXT J
850 REM DEGREES OF FREEDOM
860 V = N - K - 1
870 VTAB 22: HTAB 1: PRINT SPC( 39)
880 IF V < 1 THEN VTAB 22: HTAB 4: PRINT
"You have ";V;" degrees of freedom !";
CHR$ (7);: GOTO 770
890 RETURN
900 REM X'S
910 FOR I = 1 TO K
920 HOME
930 PRINT "Please enter observations for "
;V$(I);". "; CHR$ (7)
940 FOR J = 1 TO N
950 VTAB 3: HTAB 13: PRINT SPC( 20)
960 VTAB 3: HTAB 1: PRINT V$(I); TAB( 4)"(
";J; TAB( 9)"")= ";
970 INPUT X$
980 X(J,I) = VAL (X$)
990 NEXT J,I
1000 RETURN
1010 REM CORRECT DATA
1020 FOR I = 0 TO K
1030 FOR J = 1 TO N STEP 10
1040 REM DISPLAY
1050 GOSUB 1100
1060 REM CORRECT
1070 GOSUB 1230

```



```
1080 NEXT J,I
1090 RETURN
1100 REM DISPLAY DATA
1110 HOME
1120 PRINT L$
1130 A$ = CHR$ (32)
1140 IF J > 10 THEN A$ = " more "
1150 PRINT "These are";A$;"values of ";V$(
I);":"
1160 PRINT L$: PRINT
1170 R = 0
1180 FOR L = J TO J + 9
1190 IF L < = N THEN R = R + 1: INVERSE :
PRINT CHR$ (R + 64);: NORMAL : PRINT
" No. ";L; TAB( 10)"= ";X(L,I)
1200 NEXT L
1210 VTAB 16: HTAB 1: PRINT L$
1220 RETURN
1230 REM CORRECT DATA
1240 VTAB 21: HTAB 1: PRINT SPC( 39)
1250 VTAB 23: HTAB 1: PRINT SPC( 39)
1260 VTAB 19: HTAB 1: PRINT "Corrections (
Y/N) ? "; CHR$ (7);
1270 GET S$
1280 IF S$ = "N" OR S$ = "n" THEN 1410
1290 IF S$ < > "Y" AND S$ < > "y" THEN 1
260
1300 VTAB 21: HTAB 1: PRINT "Which letter
? "; CHR$ (7);
1310 GET S$
1320 A = ASC (S$)
1330 IF A > 90 THEN A = A - 32
1340 Q = A - 64
1350 IF Q < 1 OR Q > R THEN 1300
1360 VTAB (A - 60): HTAB 1: FLASH : PRINT
CHR$ (A); CHR$ (7);: NORMAL
1370 VTAB 23: HTAB 1: INPUT "What should t
he value be ? ";S$
1380 X(J + Q - 1,I) = VAL (S$)
1390 VTAB (A - 60): HTAB 12: PRINT SPC( 2
0);: HTAB 12: PRINT X(J + Q - 1,I)
1400 VTAB (A - 60): HTAB 1: INVERSE : PRINT
CHR$ (A);: NORMAL : GOTO 1240
1410 RETURN
1420 REM COMPUTE
1430 HOME
1440 VTAB 12: HTAB 15: FLASH : PRINT "COMP
UTING";: NORMAL
1450 REM PERFORM ORTHOGONALIZATION
1460 GOSUB 1540
```

```
1470 REM BACK-SOLVE FOR COEFFICIENTS
1480 GOSUB 2130
1490 REM COMPUTE VARIANCE-COVARIANCE MATR
IX
1500 GOSUB 2250
1510 REM TALLY SUMMARY TERMS
1520 GOSUB 2700
1530 RETURN
1540 REM ORTHOGONALIZATION
1550 REM INSERT CONSTANT TERM
1560 GOSUB 1710
1570 K = K + 1
1580 FOR Z = 1 TO K
1590 REM COMPUTE KEY ELEMENT OF R
1600 GOSUB 1800
1610 REM COMPUTE COLUMN OF Q
1620 GOSUB 1870
1630 REM COMPUTE COLUMN OF R
1640 GOSUB 1920
1650 REM COMPUTE ELEMENT OF C
1660 GOSUB 2000
1670 REM REVISE X
1680 GOSUB 2060
1690 NEXT Z
1700 RETURN
1710 REM INSERT CONSTANT
1720 FOR I = K TO 1 STEP - 1
1730 FOR J = 1 TO N
1740  $X(J, I + 1) = X(J, I)$ 
1750 NEXT J, I
1760 FOR J = 1 TO N
1770  $X(J, 1) = 1$ 
1780 NEXT J
1790 RETURN
1800 REM KEY ELEMENT OF R
1810 S = 0
1820 FOR I = 1 TO N
1830  $S = S + X(I, Z) ^ 2$ 
1840 NEXT I
1850  $R(Z, Z) = \text{SQR}(S)$ 
1860 RETURN
1870 REM COLUMN OF Q
1880 FOR I = 1 TO N
1890  $Q(I, Z) = X(I, Z) / R(Z, Z)$ 
1900 NEXT I
1910 RETURN
1920 REM COLUMN OF R
1930 IF Z = K THEN 1990
1940 FOR L = Z + 1 TO K
```

```
1950 R(Z,L) = 0
1960 FOR I = 1 TO N
1970 R(Z,L) = R(Z,L) + X(I,L) * Q(I,Z)
1980 NEXT I,L
1990 RETURN
2000 REM ELEMENT OF C
2010 C(Z) = 0
2020 FOR I = 1 TO N
2030 C(Z) = C(Z) + X(I,0) * Q(I,Z)
2040 NEXT I
2050 RETURN
2060 REM REVISE X
2070 IF Z = K THEN 2120
2080 FOR I = 1 TO N
2090 FOR L = Z + 1 TO K
2100 X(I,L) = X(I,L) - Q(I,Z) * R(Z,L)
2110 NEXT L,I
2120 RETURN
2130 REM BACK-SOLVE
2140 B(K) = C(K) / R(K,K)
2150 FOR I = K - 1 TO 1 STEP - 1
2160 REM LEFT-SIDE SUM
2170 S = 0
2180 FOR J = I + 1 TO K
2190 S = S + R(I,J) * B(J)
2200 NEXT J
2210 REM COEFFICIENT
2220 B(I) = (C(I) - S) / R(I,I)
2230 NEXT I
2240 RETURN
2250 REM VARIANCE-COVARIANCE MATRIX
2260 REM COMPUTE ERROR VARIANCE
2270 GOSUB 2330
2280 REM INVERT MATRIX R
2290 GOSUB 2490
2300 REM COMPUTE UNSCALED VAR-COV MATRIX
2310 GOSUB 2620
2320 RETURN
2330 REM ERROR VARIANCE
2340 REM RESIDUALS = Y-Q*C
2350 FOR I = 1 TO N
2360 S = 0
2370 FOR J = 1 TO K
2380 S = S + Q(I,J) * C(J)
2390 NEXT J
2400 Q(I,0) = X(I,0) - S
2410 NEXT I
2420 REM ERROR VARIANCE
2430 ESS = 0
```

```
2440 FOR I = 1 TO N
2450 ESS = ESS + Q(I,0) ^ 2
2460 NEXT I
2470 EV = ESS / V
2480 RETURN
2490 REM INVERT R
2500 FOR I = 1 TO K
2510 V(I,I) = 1 / R(I,I)
2520 NEXT I
2530 FOR I = K - 1 TO 1 STEP - 1
2540 FOR J = I + 1 TO K
2550 S = 0
2560 FOR L = I + 1 TO J
2570 S = S + R(I,L) * V(L,J)
2580 NEXT L
2590 V(I,J) = - S / R(I,I)
2600 NEXT J,I
2610 RETURN
2620 REM UNSCALED MATRIX
2630 FOR I = 1 TO K
2640 FOR J = 1 TO K
2650 R(I,J) = 0
2660 FOR L = 1 TO K
2670 R(I,J) = R(I,J) + V(I,L) * V(J,L)
2680 NEXT L,J,I
2690 RETURN
2700 REM SUMMARY STATISTICS
2710 REM SUMS OF SQUARES
2720 GOSUB 2800
2730 REM GOODNESS-OF-FIT
2740 GOSUB 2910
2750 REM DW STATISTIC
2760 GOSUB 2960
2770 REM RHO
2780 GOSUB 3030
2790 RETURN
2800 REM SUMS OF SQUARES
2810 REM TOTAL
2820 S = 0:SS = 0
2830 FOR I = 1 TO N
2840 S = S + X(I,0)
2850 SS = SS + X(I,0) ^ 2
2860 NEXT I
2870 TSS = SS - S * S / N
2880 REM REGRESSION
2890 RSS = TSS - ESS
2900 RETURN
2910 REM GOODNESS-OF-FIT STATISTICS
2920 RQ = RSS / TSS
2930 RBSQ = 1 - EV / (TSS / (N - 1))
```

```
2940 F = RSS / (K - 1) / EV
2950 RETURN
2960 REM DW STATISTIC
2970 S = 0
2980 FOR I = 2 TO N
2990 S = S + (Q(I,0) - Q(I - 1,0)) ^ 2
3000 NEXT I
3010 DW = S / ESS
3020 RETURN
3030 REM RHO
3040 REM NUMERATOR
3050 S = 0
3060 FOR I = 2 TO N
3070 S = S + Q(I,0) * Q(I - 1,0)
3080 NEXT I
3090 REM DENOMINATOR
3100 D = 0
3110 FOR I = 2 TO N - 1
3120 D = D + Q(I,0) ^ 2
3130 NEXT I
3140 RHO = S / D
3150 RETURN
3160 REM DISPLAY RESULTS
3170 REM EQUATION
3180 GOSUB 3220
3190 REM SUMMARY VALUES
3200 GOSUB 3390
3210 RETURN
3220 REM EQUATION
3230 HOME
3240 PRINT L$
3250 PRINT TAB( 11)"REGRESSION EQUATION
3260 PRINT L$
3270 VTAB 5: HTAB 15: PRINT "Estimated
3280 VTAB 6: HTAB 1: PRINT "Term"; TAB( 17
) "Value"; TAB( 33)"t-Ratio
3290 FOR I = 1 TO K
3300 T = B(I) / SQR (EV * R(I,I))
3310 VTAB I + 7: HTAB 1: PRINT B$(I - 1)
3320 VTAB I + 7: HT = 23: NR = B(I): GOSUB 9
000
3330 VTAB I + 7: HT = 39: NR = T: GOSUB 9000

3340 NEXT I
3350 VTAB 21: HTAB 1: PRINT L$
3360 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
3370 GET S$
3380 RETURN
3390 REM SUMMARY VALUES
```

```
3400 REM SUMS OF SQUARES
3410 GOSUB 3450
3420 REM KEY TERMS
3430 GOSUB 3600
3440 RETURN
3450 REM SUMS OF SQUARES
3460 HOME
3470 PRINT L$
3480 PRINT TAB( 13)"SUMMARY VALUES
3490 PRINT L$
3500 PRINT TAB( 33)"Degrees
3510 PRINT "Variation"; TAB( 20)"Sums of";
    TAB( 35)"of
3520 PRINT TAB( 3)"in Y"; TAB( 20)"Square
    s"; TAB( 33)"Freedom
3530 VTAB 8: HTAB 1: PRINT "Total";:HT = 2
    6:NR = TSS: GOSUB 9000
3540 VTAB 8:HT = 37:DA = 0:NR = N - 1: GOSUB
    9000:DA = 3
3550 VTAB 9: HTAB 1: PRINT "Regression";:H
    T = 26:NR = RSS: GOSUB 9000
3560 VTAB 9:HT = 37:DA = 0:NR = K - 1: GOSUB
    9000:DA = 3
3570 VTAB 10: HTAB 1: PRINT "Residual";:HT
    = 26:NR = ESS: GOSUB 9000
3580 VTAB 10:HT = 37:DA = 0:NR = V: GOSUB
    9000:DA = 3
3590 RETURN
3600 REM KEY TERMS
3610 VTAB 13: HTAB 1: PRINT "R-Squared"; TAB(
    17)"=";:HT = 32:NR = RQ: GOSUB 9000
3620 VTAB 14: HTAB 1: PRINT "R-Bar Squared
    "; TAB( 17)"=";:NR = RBSQ: GOSUB 9000
3630 VTAB 16: HTAB 1: PRINT "F-Statistic";
    TAB( 17)"=";:NR = F: GOSUB 9000
3640 VTAB 18: HTAB 1: PRINT "Standard Erro
    r
3650 VTAB 19: HTAB 1: PRINT "of the Estima
    te"; TAB( 17)"=";:NR = SQR (EV): GOSUB
    9000
3660 VTAB 21: HTAB 1: PRINT "Durbin-Watson
    "; TAB( 17)"=";:NR = DW: GOSUB 9000
3670 VTAB 22: HTAB 1: PRINT "Rho"; TAB( 17
    )"=";:NR = RHO: GOSUB 9000
3680 VTAB 23: HTAB 1: PRINT L$
3690 VTAB 24: HTAB 14: PRINT "Press any ke
    y ";
3700 GET S$
3710 RETURN
```

```

9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
    00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
    1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
    L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
    Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

General-Form Curve Fitter

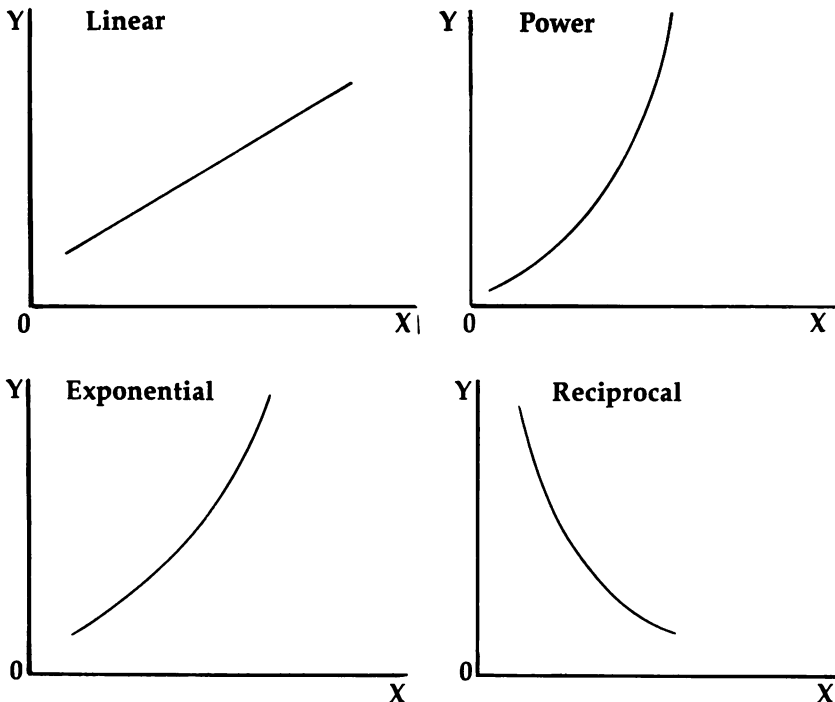
The relationship between one variable and another is not always linear. Recognizing this, "General-Form Curve Fitter" lets you estimate a simple regression equation using one of four forms.

Many times you'll want to investigate the relationship between two variables, but not in a linear fashion. By using this program, you can estimate relationships in any of four different forms:

- Linear
- Power
- Exponential
- Reciprocal

Graphed, these relationships appear as diagrammed in Figure 1.

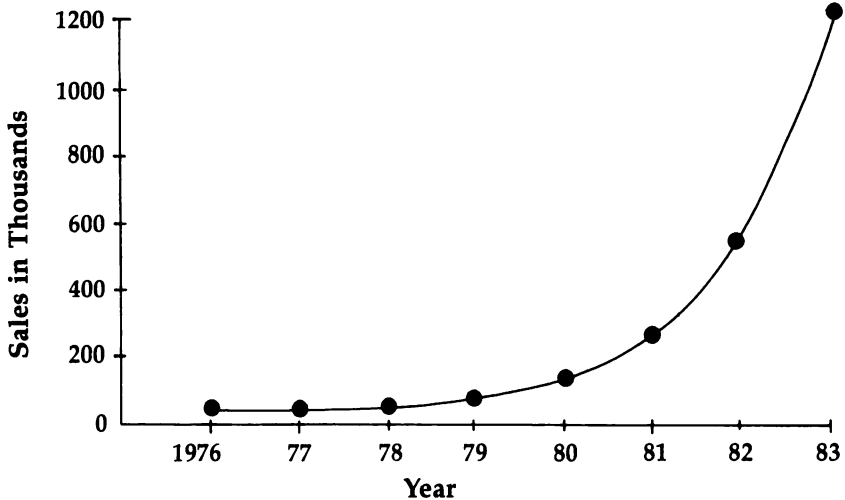
Figure 1. Variable Relationships



Exponential Example

Once you've entered, saved, and run this program, you can enter your own data. An example might be a good place to start. Suppose that you want to estimate the relationship between computer sales and time. As Figure 2 shows, sales seem to be climbing exponentially. Hence, instead of using the traditional linear equation, you instruct the computer to use the exponential equation instead.

Figure 2. Growth in Sales



The exact figures for each year are listed in the table. The numbers in the "Units Sold" and "Time" columns are the values you'll actually type into the computer.

Computer Sales

Year	Units Sold in Thousands	Time
1976	1.0	1
1977	1.5	2
1978	4.0	3
1979	10.0	4
1980	40.5	5
1981	200.0	6
1982	500.0	7
1983	1200.0	8

After entering the data from the table, with Y equal to sales and X equal to time, the computer asks which equation you want to use. Since sales appear to be exponential, press the 3 key. The computer estimates the equation and displays the regression results shown in Figure 3.

Figure 3. Regression Results

Regression Results		
For $Y = a \cdot e^{b \cdot X}$		
Term	Estimated Value	T-Ratio
Constant (a)	0.198	-5.817
Coefficient (b)	1.093	19.835
R-Squared	=	0.985
F-Statistic	=	393.436
Standard Error of the Estimate	=	0.357

Press any key

All these statistics are similar to those you've seen in the previous three articles. Refer to those articles for more complete explanations. Note, however, the high value of R-squared. This means that the exponential equation fits the data very well, explaining 98.5 percent of the variation in sales.

After digesting these results, you can estimate another regression equation using the same set of data, perhaps with a power function this time (by pressing any key, followed by pressing the 2 key). Notice the R-squared value which results. The 0.856 value for R-squared is lower than the value obtained with the exponential equation. In other words, it's a less accurate method of estimating the relationship between the variables.

When you're ready to move on to another set of data or to exit the program entirely, press the 5 key from the main menu.

Technical Notes

"General-Form Curve Fitter" estimates any of the following equations:

1. $Y = a + bX$ **Linear**
2. $Y = aX^b$ **Power**
3. $Y = a(e)^{bX}$ **Exponential**
4. $Y = a + b/X$ **Reciprocal**

It does this by using the reliable simple linear regression algorithm. The trick, of course, is transforming the power, exponential, and reciprocal functions into linear form.

In the first two cases (equations 2 and 3), this is done simply by computing logarithms to base e in this way:

$$\ln(Y) = \ln(a) + b \cdot \ln(X) \quad \text{Power}$$

$$\ln(Y) = \ln(a) + bX \quad \text{Exponential}$$

Since logarithms don't exist for negative numbers or for zero, the computer balks if it encounters either, and returns to the equation-selection part of the program after letting you know what the problem is.

When the power and exponential equations are computed, the estimated value of a is displayed, instead of $\ln(a)$. This is the reason that the mathematical signs of a and its corresponding t-ratio will sometimes differ (the t-ratio is based on $\ln(a)$, as statistical theory demands).

Finally, when the reciprocal equation is estimated, values of X are divided into one to get the transformed variable. Division by zero is checked for and avoided.

General-Form Curve Fitter

```
100 REM APPLE CURVE FITTER
110 REM INITIALIZE
120 GOSUB 290
130 REM ENTER DATA
140 GOSUB 690
150 REM EDIT DATA
160 GOSUB 1030
170 REM SELECT FUNCTION
180 GOSUB 1440
190 IF PICK = 5 THEN 280
200 REM TRANSFORM DATA
210 GOSUB 1540
220 REM COMPUTE
230 IF D$ = "BAD DATA" THEN 180
240 IF D$ = "GOOD DATA" THEN GOSUB 2060
250 REM DISPLAY RESULTS
260 GOSUB 2490
270 GOTO 180
280 END
290 REM INITIALIZE
300 REM HEADING
310 GOSUB 350
320 REM MAXIMUM NUMBER OF OBSERVATIONS
330 GOSUB 560
340 RETURN
```

```
350 REM HEADING
360 PRINT CHR$(21): TEXT : HOME
370 PRINT "This program estimates a regres
    sion
380 PRINT "equation using any of these for
    ms
390 GOSUB 460
400 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT
410 REM DIGITS AFTER DECIMAL
420 DA = 3
430 VTAB 23: HTAB 14: PRINT "Press any key
    ";
440 GET S$
450 RETURN
460 REM FUNCTIONAL FORMS
470 VTAB 5: HTAB 5: INVERSE : PRINT "EQUAT
    ION"; HTAB (26): PRINT "FORM
480 VTAB 7: HTAB 3: INVERSE : PRINT "1";: NORMAL
485 PRINT "Y = a + b*x"; TAB( 24)"Linear
490 VTAB 10: HTAB 3: INVERSE : PRINT "2";:
    NORMAL : PRINT " Y = a*X"; TAB( 24)"P
    ower
500 VTAB 9: HTAB 12: PRINT "b"
510 VTAB 13: HTAB 3: INVERSE : PRINT "3";:
    NORMAL : PRINT " Y = a*e"; TAB( 24)"E
    xponential
520 VTAB 12: HTAB 12: PRINT "b*x
530 VTAB 16: HTAB 3: INVERSE : PRINT "4";:
    NORMAL : PRINT " Y = a + b/X"; TAB( 2
    4)"Reciprocal
540 VTAB 19: HTAB 3: INVERSE : PRINT "5";:
    NORMAL : HTAB 24: PRINT "None
550 RETURN
560 REM MAXIMUM NUMBER OF OBSERVATIONS
570 HOME
580 DATA 301
590 READ NX
600 DIM X(NX,2),XT(NX),YT(NX)
610 V$(1) = "Y":V$(2) = "X"
620 PRINT "The maximum number of observati
    ons
630 PRINT "allowed is ";NX; "."
640 PRINT
650 PRINT "Change line 580 for a different
    value.
660 VTAB 23: HTAB 14: PRINT "Press any key
    ";
670 GET S$
680 RETURN
```

```
690 REM ENTER DATA
700 REM ON Y
710 GOSUB 750
720 REM ON X
730 GOSUB 920
740 RETURN
750 REM Y
760 HOME
770 PRINT "Please enter observations on th
e
780 PRINT "dependent variable, Y. Hit the
RETURN
790 PRINT "key when through."; CHR$ (7)
800 N = NX
810 FOR J = 1 TO NX
820 VTAB 5: HTAB 11: PRINT SPC( 20)
830 VTAB 5: HTAB 1: PRINT "Y( ";J; TAB( 7)
;")= ";
840 INPUT V$
850 IF V$ = "" THEN N = J - 1:J = NX
860 IF V$ < > "" THEN X(J,1) = VAL (V$)
870 NEXT J
880 REM DEGREES OF FREEDOM
890 V = N - 2
900 IF V < 1 THEN VTAB 22: HTAB 1: PRINT
"You need at least three observations
!"; CHR$ (7);: GOTO 800
910 RETURN
920 REM X
930 HOME
940 PRINT "Please enter data on the explan
atory
950 PRINT "variable, X."; CHR$ (7)
960 FOR J = 1 TO N
970 VTAB 4: HTAB 11: PRINT SPC( 20)
980 VTAB 4: HTAB 1: PRINT "X( ";J; TAB( 7)
;")= ";
990 INPUT X$
1000 X(J,2) = VAL (X$)
1010 NEXT J
1020 RETURN
1030 REM CORRECT DATA
1040 FOR I = 1 TO 2
1050 FOR J = 1 TO N STEP 10
1060 REM DISPLAY
1070 GOSUB 1120
1080 REM CORRECT
1090 GOSUB 1250
1100 NEXT J, I
1110 RETURN
```

```
1120 REM DISPLAY DATA
1130 HOME
1140 PRINT L$
1150 A$ = CHR$ (32)
1160 IF J > 10 THEN A$ = " more "
1170 PRINT "These are";A$;"values of ";V$(
I);":"
1180 PRINT L$: PRINT
1190 R = 0
1200 FOR L = J TO J + 9
1210 IF L < = N THEN R = R + 1: INVERSE :
PRINT CHR$ (R + 64);: NORMAL : PRINT
" No. ";L; TAB( 10)"= ";X(L,I)
1220 NEXT L
1230 VTAB 16: HTAB 1: PRINT L$
1240 RETURN
1250 REM CORRECT DATA
1260 VTAB 21: HTAB 1: PRINT SPC( 39)
1270 VTAB 23: HTAB 1: PRINT SPC( 39)
1280 VTAB 19: HTAB 1: PRINT "Corrections (
Y/N) ? "; CHR$ (7);
1290 GET S$
1300 IF S$ = "N" OR S$ = "n" THEN 1430
1310 IF S$ < > "Y" AND S$ < > "y" THEN 1
280
1320 VTAB 21: HTAB 1: PRINT "Which letter
? "; CHR$ (7);
1330 GET S$
1340 A = ASC (S$)
1350 IF A > 90 THEN A = A - 32
1360 Q = A - 64
1370 IF Q < 1 OR Q > R THEN 1320
1380 VTAB (A - 60): HTAB 1: FLASH : PRINT
CHR$ (A); CHR$ (7);: NORMAL
1390 VTAB 23: HTAB 1: INPUT "What should t
he value be ? ";S$
1400 X(J + Q - 1,I) = VAL (S$)
1410 VTAB (A - 60): HTAB 12: PRINT SPC( 2
0);: HTAB 12: PRINT X(J + Q - 1,I)
1420 VTAB (A - 60): HTAB 1: INVERSE : PRINT
CHR$ (A);: NORMAL : GOTO 1260
1430 RETURN
1440 REM SELECT FUNCTION
1450 HOME
1460 PRINT "Please enter the type of equat
ion that
1470 PRINT "you'd like to estimate.
1480 GOSUB 460
1490 VTAB 23: HTAB 3: PRINT "Choice (1 to
5) = ? "; CHR$ (7);
```

```
1500 GET S$
1510 PICK = VAL (S$)
1520 IF PICK < 1 OR PICK > 5 THEN 1490
1530 RETURN
1540 REM TRANSFORM DATA
1550 HOME
1560 VTAB 12: HTAB 15: FLASH : PRINT "ESTI
MATING";: NORMAL
1570 D$ = "GOOD DATA"
1580 ON PICK GOSUB 1600,1660,1840,1910
1590 RETURN
1600 REM LINEAR FUNCTION
1610 FOR I = 1 TO N
1620 YT(I) = X(I,1)
1630 XT(I) = X(I,2)
1640 NEXT I
1650 RETURN
1660 REM POWER FUNCTION
1670 FOR I = 1 TO N
1680 IF X(I,1) > 0 THEN YT(I) = LOG (X(I,
1))
1690 IF X(I,1) < = 0 THEN SYMBOL$ = "Y":I
NDEX = I:C = X(I,1): GOSUB 1740:I = N:
GOTO 1720
1700 IF X(I,2) > 0 THEN XT(I) = LOG (X(I,
2))
1710 IF X(I,2) < = 0 THEN SYMBOL$ = "X":I
NDEX = I:C = X(I,2): GOSUB 1740:I = N
1720 NEXT I
1730 RETURN
1740 REM BAD DATA
1750 HOME
1760 PRINT "Sorry, I can't estimate your e
quation.
1770 S$ = SYMBOL$ + "(" + STR$ (INDEX) + "
)"
1780 PRINT : PRINT "LOG " + CHR$ (91) + S
$ + CHR$ (93);" is undefined since
1790 PRINT : PRINT TAB( 10)S$;" = ";C;".
"
1800 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
1810 D$ = "BAD DATA"
1820 GET S$
1830 RETURN
1840 REM EXPONENTIAL FUNCTION
1850 FOR I = 1 TO N
1860 XT(I) = X(I,2)
1870 IF X(I,1) > 0 THEN YT(I) = LOG (X(I,
1))
```

```
1880 IF X(I,1) < = 0 THEN SYMBOL$ = "Y":I
      NDEX = I:C = X(I,1): GOSUB 1740:I = N
1890 NEXT I
1900 RETURN
1910 REM RECIPROCAL FUNCTION
1920 FOR I = 1 TO N
1930 YT(I) = X(I,1)
1940 IF X(I,2) < > 0 THEN XT(I) = 1 / X(I
      ,2)
1950 IF X(I,2) = 0 THEN INDEX = I: GOSUB 1
      980:I = N
1960 NEXT I
1970 RETURN
1980 REM DIVISION BY ZERO
1990 HOME
2000 PRINT "Sorry, I can't estimate your e
      quation.
2010 PRINT "X(";INDEX;") = 0, and I can't
      divide.
2020 D$ = "BAD DATA"
2030 VTAB 23: HTAB 14: PRINT "Press any ke
      y ";
2040 GET S$
2050 RETURN
2060 REM COMPUTE
2070 REM KEY SUMS
2080 GOSUB 2180
2090 REM COEFFICIENT & CONSTANT
2100 GOSUB 2280
2110 REM ANOVA TERMS
2120 GOSUB 2320
2130 REM t-STATISTICS
2140 GOSUB 2450
2150 REM TRANSFORM A & B BACK TO NON-LOG
      FORM
2160 IF PICK = 2 OR PICK = 3 THEN A = EXP
      (A)
2170 RETURN
2180 REM KEY SUMS
2190 SX = 0:SY = 0:XQ = 0:YQ = 0:CP = 0
2200 FOR I = 1 TO N
2210 SX = SX + XT(I)
2220 SY = SY + YT(I)
2230 XQ = XQ + XT(I) ^ 2
2240 YQ = YQ + YT(I) ^ 2
2250 CP = CP + XT(I) * YT(I)
2260 NEXT
2270 RETURN
2280 REM COEFFICIENT & CONSTANT
```



```
2290 B = (N * CP - SX * SY) / (N * XQ - SX *
    SX)
2300 A = (SY - B * SX) / N
2310 RETURN
2320 REM ANOVA TERMS
2330 TSS = YQ - SY * SY / N
2340 RSS = B * (CP - SX * SY / N)
2350 ESS = TSS - RSS
2360 RQ = RSS / TSS
2370 REM ERROR VARIANCE, F, & STANDARD ER
    ROR OF THE ESTIMATE
2380 EV = ESS / V
2390 F = RSS / EV
2400 SEE = SQR (EV)
2410 REM STANDARD ERRORS OF THE COEFFICIE
    NT & CONSTANT
2420 SB = SQR (EV / (XQ - SX * SX / N))
2430 SA = SQR (EV * XQ / (N * XQ - SX * SX
    ))
2440 RETURN
2450 REM t-STATISTICS
2460 TB = B / SB
2470 TA = A / SA
2480 RETURN
2490 REM DISPLAY RESULTS
2500 REM HEADING
2510 GOSUB 2580
2520 REM VALUES
2530 GOSUB 2730
2540 VTAB 23: HTAB 1: PRINT L$
2550 VTAB 24: HTAB 14: PRINT "Press any ke
    y ";
2560 GET S$
2570 RETURN
2580 REM HEADING
2590 HOME
2600 PRINT L$
2610 PRINT TAB( 11)"REGRESSION RESULTS
2620 PRINT L$
2630 S1$ = ""
2640 IF PICK = 1 THEN S2$ = "a + b*X"
2650 IF PICK = 2 THEN S2$ = "a*X":S1$ = "b
    "
2660 IF PICK = 3 THEN S2$ = "a*e":S1$ = "b
    *X"
2670 IF PICK = 4 THEN S2$ = "a + b/X"
2680 VTAB 5: HTAB 12: PRINT S1$
2690 VTAB 6: HTAB 1: PRINT "For Y = ";S2$;
    TAB( 17);": "
```

```

2700 INVERSE : VTAB 8: HTAB 15: PRINT "EST
IMATED
2710 VTAB 9: HTAB 1: PRINT "TERM";: HTAB 1
7: PRINT "VALUE";: HTAB 33: PRINT "T-R
ATIO": NORMAL
2720 RETURN
2730 REM VALUES
2740 VTAB 11: HTAB 1: PRINT "a";:HT = 23:N
R = A: GOSUB 9000
2750 VTAB 11:HT = 39:NR = TA: GOSUB 9000
2760 VTAB 12: HTAB 1: PRINT "b";:HT = 23:N
R = B: GOSUB 9000
2770 VTAB 12:HT = 39:NR = TB: GOSUB 9000
2780 VTAB 14: HTAB 1: PRINT "R-Squared"; TAB(
17)"=";:HT = 31:NR = RQ: GOSUB 9000
2790 VTAB 16: HTAB 1: PRINT "F-Statistic";
TAB( 17)"=";:NR = F: GOSUB 9000
2800 VTAB 18: HTAB 1: PRINT "Standard Erro
r
2810 VTAB 19: HTAB 1: PRINT "of the Estima
te"; TAB( 17)"=";:NR = SEE: GOSUB 9000

2820 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
"E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

5 Basics for Business



Basics for Business

Modern management techniques are often helpful in solving business problems. That's fortunate, for with volatile inflation, high interest rates, and stiff foreign competition, the business world is only for those fit to survive.

This chapter presents a series of programs designed to help corporate executives, small-company planners, and home-based entrepreneurs better manage their financial resources. Indeed, since scarcity of goods and services is a problem faced by all, you may want to use some of these routines to help manage your personal finances, too. The six programs are:

- **Database Management.** Create, update, and access a database.
- **Net Present Value.** Determine how much a future stream of dollars is worth today.
- **Internal Rate of Return.** Evaluate investment proposals.
- **Least-Squares Forecasting.** Forecast the future value of a variable through use of simple linear regression analysis.
- **Time-Series Forecasting.** Estimate the future value of a variable through use of a host of extrapolation techniques.
- **Computer Cash Register.** Turn your computer into a money-counting machine.

A Note on Numbers

Anytime you work with measurement and numbers, you should keep in mind the possible sources of error. Today's home computers are generally very accurate, but are on occasion subject to rounding errors. Use common sense when interpreting the results obtained from these programs. If the answer to a derivative calculation is 1000.0001, you should consider rounding this to 1000 if you use it in another calculation. The difference is most likely insignificant, since it is smaller than the possible error in the calculation.

1

2

3

4

5

6

7

8

9

10

Database Management

This simple program enables you to create, update, and access a database file. In a world of paper, this can be a tremendous help. No more sorting index cards, shuffling forms, or pencil-editing columns of numbers. Do it all on a disk!

Keeping track of figures is easy with this program. Whether it's your stock portfolio or your business inventory, "Database Management" makes it simple to keep track of your numbers.

Stock Portfolio

The simplest way to show the mechanics of Database Management is to run through an example. Make sure you have the program typed in, saved, and run.

Suppose you want to keep track of your stock market investments. Your current portfolio is shown in the table below.

Portfolio

Stock	ID	No. of Shares	Month Purchased	Year Purchased	Orig Price	Current Price	Dividend
IBM	1	2	2	83	109.750	115.000	3.80
Tandy	2	10	2	83	50.500	31.875	0
Marriott	3	10	2	83	54.250	62.375	0.44
USGyps	4	6	4	84	40.000	54.000	2.60
USSteel	5	6	4	84	28.375	25.125	1.00
HazLab	6	15	5	84	10.625	17.000	0.32
LILCO	7	15	6	84	10.125	11.500	2.20
Alcoa	8	3	6	84	42.000	43.500	1.20
Clorox	9	3	8	84	23.125	25.875	1.04
GnDyn	10	5	8	84	49.875	61.750	1.00

When you first run Database Management, the main menu displays. Options include creating, updating, or accessing a file, as well as exiting the program. Press the 1 key to signify that you want to create a database file.

The program then asks you how many variables and observations you want to include. After looking again at the table, enter 7 for the number of variables (the seven categories, from ID through Dividend, running across the top of the table), and 10 for the number of observations (the ten

stocks, from IBM to GnDyn, running down the left side of the table). You also need to name each variable. Since you're restricted to seven characters, you might use the following titles:

ID	Identification number
Shares	Number of shares
Month	Month of purchase
Year	Year of purchase
Price	Original price
Kprice	Current price
Div	Dividend per share

Now enter the data you see in the table. You'd type 1 for the first observation of variable ID, 2 for the second observation, and so on. The variable *Shares* would have 2 as its first observation, 10 as its second and third, 6 as its fourth and fifth, and so on.

When you've finished entering the database information, type in a filename when the prompt appears. Perhaps you'll call your file STOCKS. The computer saves the file to disk and returns you to the main menu.

Accessing, Sorting, and Updating

To access a database file already created and saved on disk, enter the appropriate filename (for example, STOCKS). The computer retrieves the information and displays a menu similar to the figure.

File Contents

File: STOCKS

Number of Variables = 7

Observations per Variable = 10

Variable Names: ID SHARES MONTH YEAR PRICE
KPRICE DIV

Would you like to

1. Sort data
2. Display data
3. Exit

Choice = ?_

You can sort or display the data, variable by variable, by pressing the 1 or 2 key, respectively. Try sorting the observa-

tions by current price, for instance; enter KPRICE for the variable name. You can then look at the sorted values by using the *Display data* option. Notice that the stocks (observations) are now listed in order of descending current price. Stock (or observation) 1, IBM, is still first, but Marriott, which was originally third, is now second.

When you want to update a database file (option 2 from the main menu), the computer shows you another display, with another menu. You can add or delete variables or observations, change a value, or exit the program. Select from it as needed. Suppose, for example, that you sell your shares of LILCO and want to strike this stock from your records. All you have to do is instruct the computer to delete the seventh observation from the file. It quickly does, and asks for any additional changes. When you're finished, press the 6 key and the file is saved again to disk. The main menu then reappears.

Use Database Management to electronically record those masses of numbers you need quick access to. It's not as powerful as the commercially available database programs, of course, but it's still fast and versatile. Best of all, it's easy to use.

Database Management

```

110 REM INITIALIZE
120 GOSUB 200
130 REM CHOOSE FROM MENU
140 GOSUB 440
150 REM MANAGE
160 ON CHOICE GOSUB 550,1260,2970
170 IF CHOICE < > 4 THEN 140
180 PRINT D$;"CLOSE" + FILE$
190 END
200 REM INITIALIZE
210 PRINT CHR$(21); TEXT : HOME
220 REM MAXIMUM NUMBERS OF VARIABLES & OB
SERVATIONS
230 DATA 10,100
240 READ KX,NX
250 DIM X(NX,KX),V$(KX)
260 REM GREETING
270 GOSUB 290
280 RETURN
290 REM GREETING
300 PRINT "This program enables you to cre
ate,
310 PRINT "update, and access a database.
320 PRINT

```

```

330 PRINT "The maximum allowable size of y
our file
340 PRINT "is
350 VTAB 8: HTAB 7: PRINT "Number of Varia
bles = ";KX
360 PRINT "Observations per Variable = ";N
X
370 VTAB 12: HTAB 1: PRINT "Change line 23
0 for different values.
380 VTAB 23: HTAB 14: PRINT "Press any key
";
390 GET S$
400 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
NEXT
410 D$ = CHR$ (4)
420 HOME
430 RETURN
440 REM MENU
450 VTAB 5: HTAB 10: PRINT "Would you like
to
460 VTAB 7: HTAB 10: INVERSE : PRINT "1";:
NORMAL : PRINT " Create a database

470 VTAB 9: HTAB 10: INVERSE : PRINT "2";:
NORMAL : PRINT " Update a database

480 VTAB 11: HTAB 10: INVERSE : PRINT "3";
: NORMAL : PRINT " Access data
490 VTAB 13: HTAB 10: INVERSE : PRINT "4";
: NORMAL : PRINT " Exit
500 VTAB 15: HTAB 10: PRINT "Choice = ? ";
CHR$ (7);
510 GET S$
520 CHOICE = VAL (S$)
530 IF CHOICE < 1 OR CHOICE > 4 THEN 500
540 RETURN
550 REM CREATE
560 REM ENTER FILE SIZE
570 GOSUB 670
580 REM NAME VARIABLES
590 GOSUB 810
600 REM ENTER DATA
610 GOSUB 920
620 REM NAME FILE
630 GOSUB 1030
640 REM WRITE
650 GOSUB 1080
660 RETURN
670 REM FILE SIZE
680 HOME

```

```

690 PRINT "How many variables do you have
700 VTAB 2: HTAB 12: PRINT SPC( 15); CHR$
    (7)
710 VTAB 2: HTAB 1: PRINT "(1 to ";KX;")";
    TAB( 11);
720 INPUT K$
730 K = INT ( VAL (K$))
740 IF K < 1 OR K > KX THEN 700
750 VTAB 4: HTAB 35: PRINT SPC( 15); CHR$
    (7)
760 VTAB 4: HTAB 1: PRINT "How many observ
    ations (1 to ";NX;")"; TAB( 34);
770 INPUT N$
780 N = INT ( VAL (N$))
790 IF N < 1 OR N > NX THEN 750
800 RETURN
810 REM NAME VARIABLES
820 HOME
830 PRINT "Please name each variable. Up t
    o seven
840 PRINT "characters are allowed.
850 FOR I = 1 TO K
860 VTAB 4: HTAB 15: PRINT SPC( 20); CHR$
    (7)
870 VTAB 4: HTAB 1: PRINT "Name No. ";I; TAB(
    12);"=" ;
880 INPUT V$(I)
890 IF V$(I) = "" OR LEN (V$(I)) > 7 THEN
    860
900 NEXT
910 RETURN
920 REM ENTER DATA
930 FOR I = 1 TO K
940 HOME
950 PRINT "Please enter observations for "
    ;V$(I);". "; CHR$ (7)
960 FOR J = 1 TO N
970 VTAB 3: HTAB 17: PRINT SPC( 15)
980 VTAB 3: HTAB 1: PRINT "Datum No. ";J; TAB(
    14);"=" ;
990 INPUT X$
1000 X(J,I) = VAL (X$)
1010 NEXT J,I
1020 RETURN
1030 REM NAME FILE
1040 HOME : PRINT CHR$ (7);
1050 INPUT "What's the name of your file ?
    ";FILE$
1060 IF FILE$ = "" THEN 1040
1070 RETURN

```

```

1080 REM WRITE
1090 HOME
1100 VTAB 12: HTAB 15: FLASH : PRINT "SAVI
      NG DATA": NORMAL
1110 REM PARAMETERS
1120 PRINT D$;"OPEN" + FILE$
1130 PRINT D$;"WRITE" + FILE$
1140 PRINT N;" ";K
1150 FOR I = 1 TO K
1160 PRINT V$(I)
1170 NEXT
1180 REM DATA
1190 FOR I = 1 TO K
1200 FOR J = 1 TO N
1210 PRINT X(J,I)
1220 NEXT J,I
1230 PRINT D$;"CLOSE" + FILE$
1240 HOME : PRINT FILE$;" is saved.
1250 RETURN
1260 REM UPDATE
1270 REM ENTER FILE NAME
1280 GOSUB 1030
1290 REM READ IN DATA
1300 GOSUB 1410
1310 REM DESCRIBE FILE
1320 GOSUB 1570
1330 REM CHOOSE FROM MENU
1340 GOSUB 1720
1350 ON PICK GOSUB 2010,2190,2500,2590,277
      0
1360 IF PICK < > 6 THEN 1320
1370 REM WRITE
1380 IF N = 0 OR K = 0 THEN HOME : PRINT
      "No data !": GOTO 1400
1390 GOSUB 1080
1400 RETURN
1410 REM READ DATA
1420 HOME
1430 VTAB 12: HTAB 14: FLASH : PRINT "READ
      ING DATA": NORMAL
1440 REM PARAMETERS
1450 PRINT D$;"OPEN" + FILE$
1460 PRINT D$;"READ" + FILE$
1470 INPUT N,K
1480 FOR I = 1 TO K
1490 INPUT V$(I)
1500 NEXT
1510 FOR I = 1 TO K
1520 FOR J = 1 TO N
1530 INPUT X(J,I)

```

```

1540 NEXT J,I
1550 PRINT D$;"CLOSE" + FILE$
1560 RETURN
1570 REM DESCRIBE FILE
1580 HOME
1590 PRINT L$
1600 PRINT "File: ";FILE$
1610 PRINT "Number of Variables"; TAB( 27)
    "= ";K
1620 PRINT "Observations per Variable = ";
    N
1630 PRINT
1640 PRINT "Variable Names: ";
1650 IF K = 0 THEN 1690
1660 FOR I = 1 TO K
1670 PRINT V$(I),
1680 NEXT
1690 VTAB 10: HTAB 1: PRINT L$
1700 VTAB 23: HTAB 1: PRINT L$
1710 RETURN
1720 REM CHOOSE
1730 VTAB 12: HTAB 10: PRINT "Would you li
    ke to
1740 INVERSE : FOR I = 1 TO 6: VTAB I + 13
    : HTAB 10: PRINT I: NEXT
1750 NORMAL
1760 VTAB 14: HTAB 12: PRINT "Add a variab
    le
1770 VTAB 15: HTAB 12: PRINT "Delete a var
    iable
1780 VTAB 16: HTAB 12: PRINT "Add an obser
    vation
1790 VTAB 17: HTAB 12: PRINT "Delete an ob
    servation
1800 VTAB 18: HTAB 12: PRINT "Change a val
    ue
1810 VTAB 19: HTAB 12: PRINT "Exit
1820 VTAB 21: HTAB 10: PRINT "Choice = ? "
    ; CHR$( 7);
1830 GET S$
1840 PICK = VAL (S$)
1850 IF PICK < 1 OR PICK > 6 THEN 1820
1860 REM CHECK RESPONSE
1870 M$ = "": GOSUB 1920
1880 VTAB 22: HTAB 1: PRINT SPC( 40);
1890 IF M$ < > "" THEN VTAB 22: HTAB (20
    - LEN (M$) / 2): PRINT M$: GOTO 1820

1900 IF PICK < > 6 THEN GOSUB 1570
1910 RETURN

```

```

1920 REM RESPONSES
1930 IF PICK = 1 AND K = KX THEN M$ = "Only " + STR$ (KX) + " variables allowed !"
1940 IF PICK = 1 AND N = 0 THEN M$ = "No observations !"
1950 IF PICK = 3 AND N = NX THEN M$ = "Only " + STR$ (NX) + " observations allowed !"
1960 IF PICK = 3 AND K = 0 THEN M$ = "No variables !"
1970 IF PICK = 2 AND K = 0 THEN M$ = "No variables !"
1980 IF PICK = 4 AND N = 0 THEN M$ = "No observations !"
1990 IF PICK = 5 AND (N = 0 OR K = 0) THEN M$ = "No data !"
2000 RETURN
2010 REM ADD A VARIABLE
2020 REM NAME
2030 K = K + 1
2040 VTAB 12: PRINT "Please name your variable. Up to seven characters are allowed."
2050 PRINT "characters are allowed."
2060 VTAB 15: HTAB 10: PRINT SPC( 20); CHR$(7)
2070 VTAB 15: HTAB 1: INPUT "Name = ? ";V$

2080 IF V$ = "" OR LEN (V$) > 7 THEN 2060
2090 V$(K) = V$
2100 REM DATA
2110 HOME
2120 PRINT "Please enter observations for ";V$
2130 FOR J = 1 TO N
2140 VTAB 3: HTAB 17: PRINT SPC( 15)
2150 VTAB 3: HTAB 1: PRINT "Datum No. ";J; TAB( 14)" = ";
2160 INPUT X$:X(J,K) = VAL (X$)
2170 NEXT
2180 RETURN
2190 REM DELETE
2200 REM NAME
2210 VTAB 12: HTAB 1: PRINT "Which variable do you want to
2220 VTAB 13: HTAB 10: PRINT SPC( 20); CHR$(7)
2230 VTAB 13: HTAB 1: INPUT "delete ? ";V$
2240 REM CHECK ENTRY
2250 GOSUB 2320

```

```

2260 IF NM = 0 THEN 2220
2270 REM DELETE
2280 IF NM < > K THEN GOSUB 2390
2290 K = K - 1
2300 IF K = 0 THEN N = 0
2310 RETURN
2320 REM CHECK
2330 NM = 0
2340 FOR I = 1 TO K
2350 IF V$ = V$(I) THEN NM = I
2360 NEXT
2370 IF NM = 0 THEN VTAB 22: HTAB 13: PRINT
    "Not on file !";
2380 RETURN
2390 REM DELETE
2400 REM NAME
2410 FOR I = NM + 1 TO K
2420 V$(I - 1) = V$(I)
2430 NEXT
2440 REM DATA
2450 FOR I = NM + 1 TO K
2460 FOR J = 1 TO N
2470 X(J,I - 1) = X(J,I)
2480 NEXT J,I
2490 RETURN
2500 REM ADD OBSERVATION
2510 VTAB 12: HTAB 1: PRINT "Please enter
    an additional observation
2520 PRINT "for each variable.
2530 N = N + 1
2540 FOR I = 1 TO K
2550 VTAB 15: HTAB 1: PRINT V$(I);"( ";N;"
    ) = ";;C = POS (0): PRINT SPC( 15):
    VTAB 15: HTAB C: INPUT X$
2560 X(N,I) = VAL (X$)
2570 NEXT
2580 RETURN
2590 REM DELETE OBSERVATION
2600 VTAB 12: HTAB 1: PRINT "Please enter
    the number of the
2610 PRINT "observation to delete (1 to ";
    N;")."
2620 VTAB 15: HTAB 12: PRINT SPC( 15); CHR$
    (7)
2630 VTAB 15: HTAB 1: INPUT "Number = ? ";
    N$
2640 ROW = INT ( VAL (N$))
2650 IF ROW < 1 OR ROW > N THEN VTAB 22: HTAB
    15: PRINT "Try again !"; GOTO 2620
2660 REM DELETE

```

```

2670 IF ROW < > N THEN GOSUB 2710
2680 N = N - 1
2690 IF N = 0 THEN K = 0
2700 RETURN
2710 REM DELETE
2720 FOR I = 1 TO K
2730 FOR J = ROW + 1 TO N
2740 X(J - 1, I) = X(J, I)
2750 NEXT J, I
2760 RETURN
2770 REM CHANGE A VALUE
2780 REM NAME
2790 VTAB 12: HTAB 1: PRINT "Please enter
the variable name and
2800 PRINT "observation number to be chang
ed.
2810 VTAB 15: HTAB 19: PRINT SPC( 20); CHR$
(7)
2820 VTAB 15: HTAB 1: INPUT "Variable Name
= ? "; V$
2830 REM CHECK ENTRY
2840 GOSUB 2320
2850 IF NM = 0 THEN 2810
2860 REM NUMBER
2870 VTAB 22: HTAB 13: PRINT SPC( 13);
2880 VTAB 17: HTAB 24: PRINT SPC( 15); CHR$
(7)
2890 VTAB 17: HTAB 1: INPUT "Observation N
umber = ? "; N$
2900 ROW = INT ( VAL (N$))
2910 IF ROW < 1 OR ROW > N THEN VTAB 22: HTAB
15: PRINT "Try again !": GOTO 2880
2920 REM VALUES
2930 VTAB 19: HTAB 1: PRINT "Old value = "
; X(ROW, NM); CHR$ (7)
2940 VTAB 20: HTAB 1: INPUT "New value = "
; X$
2950 X(ROW, NM) = VAL (X$)
2960 RETURN
2970 REM ACCESS
2980 REM ENTER FILE NAME
2990 GOSUB 1030
3000 REM READ DATA
3010 GOSUB 1410
3020 REM DESCRIBE FILE
3030 GOSUB 1570
3040 REM CHOOSE FROM MENU
3050 GOSUB 3100
3060 ON PICK GOSUB 3200, 3450
3070 IF PICK < > 3 THEN 3030

```



```

3080 HOME
3090 RETURN
3100 REM MENU
3110 VTAB 12: HTAB 10: PRINT "Would you li
ke to
3120 INVERSE : VTAB 14: HTAB 10: PRINT "1"
;: NORMAL : PRINT " Sort data
3130 INVERSE : VTAB 16: HTAB 10: PRINT "2"
;: NORMAL : PRINT " Display data
3140 INVERSE : VTAB 18: HTAB 10: PRINT "3"
;: NORMAL : PRINT " Exit
3150 VTAB 20: HTAB 10: PRINT "Choice = ? "
; CHR$ (7);
3160 GET S$
3170 PICK = VAL (S$)
3180 IF PICK < 1 OR PICK > 3 THEN 3150
3190 RETURN
3200 REM SORT
3210 GOSUB 1570
3220 VTAB 12: HTAB 1: PRINT "Which variabl
e would you like to sort
3230 VTAB 13: HTAB 6: PRINT SPC( 20); CHR$
(7)
3240 VTAB 13: HTAB 1: INPUT "on ? ";V$
3250 REM CHECK
3260 GOSUB 2320
3270 IF NM = 0 THEN 3230
3280 REM SORT
3290 GOSUB 3310
3300 RETURN
3310 REM SORT
3320 HOME
3330 VTAB 13: HTAB 16: FLASH : PRINT "SORT
ING": NORMAL
3340 SWAP$ = "NO"
3350 FOR J = 1 TO N - 1
3360 IF X(J,NM) > = X(J + 1,NM) THEN 3420

3370 FOR L = 1 TO K
3380 HD = X(J,L):X(J,L) = X(J + 1,L)
3390 X(J + 1,L) = HD
3400 NEXT L
3410 SWAP$ = "YES"
3420 NEXT J
3430 IF SWAP$ = "YES" THEN 3340
3440 RETURN
3450 REM DISPLAY DATA
3460 FOR I = 1 TO K
3470 FOR J = 1 TO N STEP 10
3480 GOSUB 3510

```

```

3490 NEXT J,I
3500 RETURN
3510 REM DISPLAY
3520 HOME
3530 PRINT L$
3540 A$ = CHR$ (32)
3550 IF J > 10 THEN A$ = " more "
3560 PRINT "These are";A$;"values of ";V$(
I);";"
3570 PRINT L$: PRINT
3580 FOR L = J TO J + 9
3590 IF L < = N THEN PRINT "Datum No. ";
L; TAB( 14)"= ";X(L,I)
3600 NEXT L
3610 VTAB 16: HTAB 1: PRINT L$
3620 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
3630 GET S$
3640 RETURN

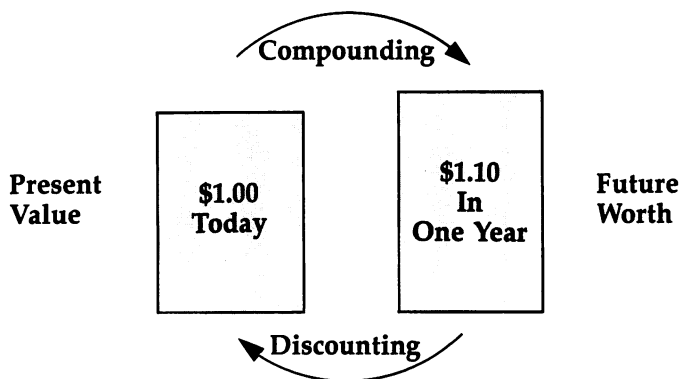
```

Net Present Value

The value of money changes as time goes by. A dollar today may not be worth a dollar a year from now. "Net Present Value" can tell you how much a future group of dollars is worth right now. This information is invaluable if you're forced to choose between investments.

You probably know firsthand the ups and downs of the dollar. For example, \$1.00 today will be worth \$1.10 in a year, assuming a 10 percent interest rate and annual compounding. On the other hand, \$1.10 a year in the future would be the same as \$1.00 now. You can call the first concept *future worth* and the second *present value*. The two, as Figure 1 shows, are related.

Figure 1. Future Worth and Present Value



We'll look at present value with this program. You might not be familiar with the concept, but it can prove valuable when you're trying to decide which of two investments is the better.

Spend or Save

Let's assume that a firm has the option of buying a new piece of equipment for \$1,500. The company expects the equipment to generate \$1,000 of revenue per annum for two years before the machinery falls apart. Alternatively, the firm can put the \$1,500 into a money market certificate and earn 10 percent interest. In this light, should the company buy the machine?

To find out, you need to type in, save, and run "Net

Present Value." You first enter 2, the number of periods in the cash flow, to represent the two years of the investment. Then key in the initial cost of the proposed investment (1500), followed by the net revenue expected in each of the two years (1000). You don't need to type in dollar signs. The computer asks for the discount rate. This percentage represents the firm's next-best rate of return on an investment (the interest rate from the money market certificate mentioned above, for example). Enter 10 for 10 percent.

The display should be similar to Figure 2.

Figure 2. Present Value

Net Present Value	
Initial cost of investment =	\$1,500.00
Number of periods =	2.00
Discount rate =	10.00%
Net Present Value =	\$ 235.54

Since the net present value is greater than zero, actually \$235.54, the company's proposed investment is economically worthwhile. Your recommendation, then, is to buy the machine.

A net present value *less* than zero implies a poor investment. In this case, the alternative (the investment represented by the discount rate) will bring a greater rate of return.

Net Present Value

```

100 REM NET PRESENT VALUE
110 REM INITIALIZE
120 GOSUB 220
130 REM ENTER INTEREST RATE
140 GOSUB 740
150 REM COMPUTE
160 GOSUB 870
170 REM DISPLAY RESULTS
180 GOSUB 990
190 REM USE ANOTHER INTEREST RATE
200 IF S$ = "Y" OR S$ = "y" THEN 140
210 END
220 REM INITIALIZE
230 REM HEADING
240 GOSUB 300
250 REM LENGTH OF CASH FLOW

```

```

260 GOSUB 430
270 REM OBSERVATIONS
280 GOSUB 560
290 RETURN
300 REM HEADING
310 PRINT CHR$(21): TEXT : HOME
320 PRINT " This program computes the Net
    Present
330 PRINT "Value of an investment.
340 PRINT
350 PRINT " Net Present Value is the amou
    nt of
360 PRINT "dollars today that will generat
    e a
370 PRINT "future cash flow, using an inte
    rest rate";
380 PRINT "of your choice.
390 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT I
400 REM DIGITS AFTER DECIMAL
410 DA = 2
420 RETURN
430 REM LENGTH OF CASH FLOW
440 REM MAXIMUM NUMBER OF PERIODS
450 DATA 100
460 READ NX
470 DIM R(NX)
480 VTAB 10: HTAB 1: PRINT "How many perio
    ds are in your
490 VTAB 11: HTAB 13: PRINT SPC(15); CHR$(
    7)
500 VTAB 11: HTAB 1: INPUT "cash flow ? ";
    N$
510 N = INT ( VAL (N$))
520 IF N < 1 THEN 490
530 IF N > NX THEN VTAB 22: HTAB 9: PRINT
    "Only ";NX;" are allowed !"
540 IF N > NX THEN VTAB 23: HTAB 4: PRINT
    "Please change line 450 for more.": GOTO
    490
550 RETURN
560 REM OBSERVATIONS
570 HOME
580 REM INITIAL COST
590 PRINT "What is the initial cost of you
    r
600 VTAB 2: HTAB 23: PRINT SPC(15); CHR$(
    7)
610 VTAB 2: HTAB 1: INPUT "proposed invest
    ment ? ";C$

```

```

620 CT = VAL (C$)
630 IF CT < 0 THEN 600
640 REM EXPECTED REVENUES
650 VTAB 5: HTAB 1: PRINT "Please enter the
net revenue (Revenue
660 PRINT "minus Cost) expected in each pe
riod of
670 PRINT "your cash flow.
680 FOR I = 1 TO N
690 VTAB 10: HTAB 14: PRINT SPC( 15)
700 VTAB 10: HTAB 1: PRINT "Period ";I; TAB(
11)"= "; INPUT V$
710 R(I) = VAL (V$)
720 NEXT
730 RETURN
740 REM INTEREST RATE
750 HOME
760 PRINT " Please enter the 'discount' o
r
770 PRINT "interest rate that you would li
ke to
780 PRINT "use in computing Net Present Va
lue.
790 PRINT
800 PRINT " For example, enter 10 for 10%
, 7 for
810 PRINT "7%, and so on.
820 VTAB 9: HTAB 14: PRINT SPC( 15); CHR$
(7)
830 VTAB 9: HTAB 5: INPUT "Rate = ? ";R$
840 RT = VAL (R$)
850 IF RT < = 0 THEN VTAB 23: HTAB 1: PRINT
"There's no such thing as a free lunch
!"; GOTO 820
860 RETURN
870 REM COMPUTE
880 HOME
890 VTAB 12: HTAB 15: FLASH : PRINT "COMPU
TING": NORMAL
900 FOR TIME = 1 TO 500: NEXT
910 INDEX = 1 + RT / 100
920 NPV = - CT
930 DF = INDEX
940 FOR I = 1 TO N
950 NPV = NPV + R(I) / DF
960 DF = DF * INDEX
970 NEXT
980 RETURN
990 REM DISPLAY
1000 HOME

```

```

1010 PRINT L$
1020 PRINT TAB( 12)"NET PRESENT VALUE
1030 PRINT L$
1040 VTAB 5: HTAB 1: PRINT "Initial Cost
1050 PRINT "of the Investment"; TAB( 19);"
    =";:HT = 36:DOLL$ = "YES":NR = CT: GOSUB 9000
9000:DOLL$ = "NO"
1060 VTAB 8: HTAB 1: PRINT "Number of Peri
    ods"; TAB( 19)"=";;:NR = N: GOSUB 9000

1070 VTAB 10: HTAB 1: PRINT "Discount Rate
    "; TAB( 19)"=";;:NR = RT: GOSUB 9000
1080 VTAB 10: HTAB 38: PRINT "%"
1090 VTAB 14: HTAB 1: PRINT "Net Present V
    alue"; TAB( 19)"=";;:DOLL$ = "YES":NR =
    NPV: GOSUB 9000:DOLL$ = "NO"
1100 VTAB 21: HTAB 1: PRINT L$
1110 VTAB 22: HTAB 4: PRINT "Use another I
    nterest Rate (Y/N) ? "; CHR$ (7);
1120 GET S$
1130 IF S$ < > "Y" AND S$ < > "y" AND S$
    < > "N" AND S$ < > "n" THEN 1110
1140 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
    00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
    1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
    L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
    Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Internal Rate of Return

Together with "Net Present Value," this program can help you decide between two different investments. It shows the rate of return you'd need to receive from one investment to make another uneconomical.

Many investments generate income for a long period, say for 10 or 20 years, while the bulk of the cost is incurred at startup time. You need to know if those dollars received in the future will be worth the initial investment, and if it will be a good investment. You can find answers to those questions with this program.

Net present value, discussed in the previous article, indicates how much a future stream of dollars is worth today. Net present value is tallied by discounting future net revenues, or revenues minus costs, using an interest rate that represents the cost of borrowing money.

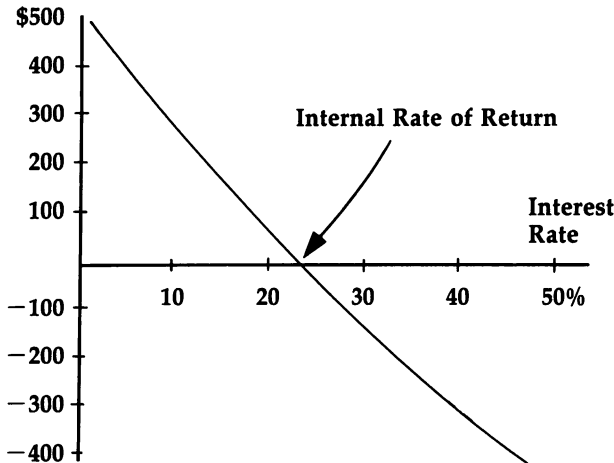
When you analyze net present value, you usually match up the proposed investment to a rate of return from another investment, perhaps a money market certificate. If the net present value is positive, the first investment is the better. But it's important to know the point of balance—at what rate of return the second investment (the money market) is the wiser choice. "Internal Rate of Return" shows you.

Spend or Save, Revisited

Let's use the example from the previous article to illustrate this. Type in, save, and run Internal Rate of Return. Enter 2 for the time periods, representing the two years of the investment. Initial cost is \$1,500 and net revenue in each year is \$1,000. If you used a money market interest rate of 10 percent as a comparison, net present value is roughly \$235. However, a 20 percent discount rate shows a net present value of only \$28. Indeed, net present value always falls when the discount rate rises, as the figure shows.

Value Versus Interest

Net Present Value



The point where the line indicating the net present value crosses the interest-rate axis is called the *internal rate of return* (IRR). That's when the net present value of a cash flow equals zero. An IRR greater than the money market rate in general means that the first investment is the better. One less than the interest rate, however, means that the first investment should be avoided.

Your computer tells you all this in a matter of minutes. It even shows you the exact interest rate you'd need in order to equal the return of the first investment. Once you've entered all the requested data (including a guess and another value called delta, which is added to or subtracted from subsequent guesses as needed), the program displays the IRR. In our example, the imaginary company would have to be getting a return of 21.53 percent on its money market certificates before the two investments are equally attractive.

Program Notes

Newton's method of discovering the root of a polynomial is used in tallying the IRR. This requires a guess of the true root, or IRR, which the program makes for you unless you alter it. The more accurate the initial guess, the faster the algorithm will find the IRR. In some cases, however, a whole series of

initial guesses (ten in all) will fail to discover the IRR. When this happens, the computer will ask you to try doubling or tripling the value of the initial guess and the increment for successive guesses.

Internal Rate of Return

```

100 REM IRR
110 REM INITIALIZE
120 GOSUB 230
130 REM ENTER DATA
140 GOSUB 670
150 REM DISPLAY INITIAL GUESS
160 GOSUB 860
170 REM COMPUTE
180 GOSUB 1020
190 REM DISPLAY RESULT
200 IF ST$ = "NON-CONVERGENCE" OR RT < 0 THEN
    GOSUB 1620: GOTO 150
210 GOSUB 1740
220 END
230 REM INITIALIZE
240 REM GREETING
250 GOSUB 310
260 REM INITIAL VALUES
270 GOSUB 430
280 REM LENGTH OF CASH FLOW
290 GOSUB 580
300 RETURN
310 REM GREETING
320 PRINT CHR$(21): TEXT : HOME
330 PRINT "This program tallies the Internal Rate
    al Rate
340 PRINT "of Return (IRR) on an investment.
350 PRINT
360 PRINT "An IRR is that interest rate which makes";
370 PRINT "the net present value of a cash
    flow
380 PRINT "equal to zero.
390 L$ = "": FOR I = 1 TO 39: L$ = L$ + "=":
    NEXT
400 REM DIGITS AFTER DECIMAL
410 DA = 2
420 RETURN
430 REM INITIAL VALUES
440 REM MAXIMUM NUMBER OF PERIODS
450 DATA 100
460 READ NX
    
```

```

470 DIM DERV(NX),POLY(NX),REVN(NX)
480 REM INITIAL GUESS OF THE IRR, & DELTA
    FOR SUCCESSIVE GUESSES
490 DATA 0.4,0.1
500 READ G0,D0
510 REM MAXIMUM NUMBER OF INITIAL GUESSES
    & ITERATIONS PER GUESS
520 DATA 10,50
530 READ M1,M2
540 REM TOLERANCE FACTOR
550 DATA 0.01
560 READ TL
570 RETURN
580 REM LENGTH
590 VTAB 8: HTAB 1: PRINT "How many period
    s are in your cash
600 VTAB 9: HTAB 8: PRINT SPC( 20); CHR$
    (7)
610 VTAB 9: HTAB 1: INPUT "flow ? ";N$
620 N = INT ( VAL (N$))
630 IF N < 1 THEN 600
640 IF N > NX THEN VTAB 20: HTAB 9: PRINT
    "Only ";NX;" are allowed !"
650 IF N > NX THEN VTAB 22: HTAB 7: PRINT
    "Change line 450 for more.": GOTO 600
660 RETURN
670 REM OBSERVATIONS
680 HOME
690 REM INITIAL COST
700 PRINT "What is the initial cost of you
    r
710 VTAB 2: HTAB 23: PRINT SPC( 15); CHR$
    (7)
720 VTAB 2: HTAB 1: INPUT "proposed invest
    ment ? ";C$
730 CT = VAL (C$)
740 IF CT < 0 THEN 710
750 REVN(0) = - CT
760 REM EXPECTED REVENUES
770 VTAB 5: HTAB 1: PRINT "Please enter th
    e net revenue (Revenue
780 PRINT "minus Cost) expected in each pe
    riod of
790 PRINT "your cash flow."; CHR$ (7)
800 FOR I = 1 TO N
810 VTAB 10: HTAB 14: PRINT SPC( 15)
820 VTAB 10: HTAB 1: PRINT "Period ";I; TAB(
    11)"= ";: INPUT V$
830 REVN(I) = VAL (V$),
840 NEXT

```

```

850 RETURN
860 REM DISPLAY INITIAL GUESSES
870 HOME :GS = G0:DL = D0
880 PRINT "The IRR is tallied using an iterative
890 PRINT "procedure. This requires a guess of
900 PRINT "the true value, shown here.
910 PRINT
920 PRINT "If you like this figure, simply hit the
930 PRINT "RETURN key. If not, enter your choice.
940 VTAB 10: HTAB 1: PRINT "Initial Guess"
; TAB( 19): ";; INVERSE : PRINT GS *
100;" %";: NORMAL
950 INPUT " ";G$
960 IF G$ < > "" THEN GS = VAL (G$) / 100
970 VTAB 12: HTAB 1: PRINT "Delta for
980 PRINT "Successive Guesses: ";; INVERSE
: PRINT DL * 100;" %";: NORMAL
990 INPUT " ";D$
1000 IF D$ < > "" THEN DL = VAL (D$) / 100
1010 RETURN
1020 REM COMPUTE
1030 HOME
1040 REM GENERATE COEFFICIENTS OF POLYNOMIAL & ITS DERIVATIVE
1050 GOSUB 1090
1060 REM TALLY IRR
1070 GOSUB 1190
1080 RETURN
1090 REM GENERATE COEFFICIENTS
1100 REM POLYNOMIAL
1110 FOR I = 0 TO N
1120 POLY(I) = REVN(N - I)
1130 NEXT
1140 REM DERIVATIVE
1150 FOR I = 0 TO N
1160 DERV(I) = I * POLY(I)
1170 NEXT
1180 RETURN
1190 REM TALLY IRR
1200 FOR I = 1 TO M1
1210 ST$ = "NON-CONVERGENCE"
1220 VTAB 13: HTAB 10: PRINT "Initial Guess = ";GS * 100;" %"

```

```

1230 REM ESTIMATE IRR ITERATIVELY
1240 GOSUB 1310
1250 REM CHECK FOR CONVERGENCE & A NON-NE
      GATIVE IRR
1260 IF ST$ = "CONVERGENCE" AND RT > = 0 THEN
      I = M1: GOTO 1290
1270 REM TRY AGAIN
1280 GS = GS + DL
1290 NEXT I
1300 RETURN
1310 REM ESTIMATE
1320 REM NEW & OLD GUESSES
1330 RT = GS
1340 HD = - 999
1350 REM MAKE UP TO M2 ESTIMATES
1360 FOR COUNT = 1 TO M2
1370 REM EVALUATE POLYNOMIAL & DERIVATIVE
      AT PRESENT ESTIMATE
1380 VTAB 15: HTAB 26: PRINT SPC( 5)
1390 VTAB 15: HTAB 14: PRINT "Iteration =
      ";COUNT
1400 GOSUB 1500
1410 REM MAKE A NEW 'INITIAL' GUESS IF DE
      RIVATIVE IS ZERO
1420 IF DV = 0 THEN COUNT = M2: GOTO 1480
1430 REM NEW ESTIMATE
1440 RT = RT - PY / DV
1450 REM CHECK FOR CONVERGENCE
1460 IF ABS (RT - HD) < TL THEN ST$ = "CO
      NVERGENCE":COUNT = M2
1470 HD = RT
1480 NEXT COUNT
1490 RETURN
1500 REM EVALUATE FUNCTIONS
1510 REM POLYNOMIAL
1520 PY = 0
1530 FOR L = 0 TO N
1540 PY = PY + POLY(L) * (1 + RT) ^ L
1550 NEXT L
1560 REM DERIVATIVE
1570 DV = 0
1580 FOR L = 1 TO N
1590 DV = DV + DERV(L) * (1 + RT) ^ (L - 1)

1600 NEXT L
1610 RETURN
1620 REM NON-CONVERGENCE
1630 HOME
1640 PRINT "Sorry, estimates of the Intern
      al Rate

```

```

1650 PRINT "of Return of your investment a
      re not
1660 PRINT "converging.
1670 PRINT
1680 PRINT "Try doubling or tripling the v
      alue of
1690 PRINT "the initial guess, and the inc
      rement
1700 PRINT "for successive guesses.
1710 VTAB 23: HTAB 14: PRINT "Press any ke
      y ";
1720 GET S$
1730 RETURN
1740 REM DISPLAY
1750 REM TALLY NET PRESENT VALUE
1760 GOSUB 1800
1770 REM RESULT
1780 GOSUB 1860
1790 RETURN
1800 REM NET PRESENT VALUE
1810 PV = 0
1820 FOR I = 0 TO N
1830 PV = PV + REVN(I) / (1 + RT) ^ I
1840 NEXT I
1850 RETURN
1860 REM RESULT
1870 HOME
1880 PRINT L$
1890 PRINT TAB( 9)"INTERNAL RATE OF RETUR
      N
1900 PRINT L$
1910 VTAB 6: HTAB 1: PRINT "Initial Cost
1920 PRINT "of the Investment"; TAB( 19)"=
      ";:DOLL$ = "YES":HT = 35:NR = CT: GOSUB
      9000
1930 VTAB 9: HTAB 1: PRINT "Number of Peri
      ods"; TAB( 19);"=";:NR = N:DOLL$ = "":
      GOSUB 9000
1940 VTAB 11: HTAB 1: PRINT "IRR"; TAB( 19
      )"=";:NR = RT * 100: GOSUB 9000
1950 VTAB 11: HTAB 37: PRINT "%"
1960 VTAB 13: HTAB 1: PRINT "Net Present V
      alue
1970 PRINT "at IRR"; TAB( 19)"=";:NR = PV:
      DOLL$ = "YES": GOSUB 9000
1980 VTAB 22: HTAB 1: PRINT L$
1990 VTAB 23: HTAB 14: PRINT "Press any ke
      y ";
2000 GET S$
2010 RETURN

```

```

9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
    00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
    1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
    L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
    Z$ + STR$ (ZR),DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Least-Squares Forecasting

The social object of skilled investment should be to defeat the dark forces of time and ignorance which envelop our future.—John Maynard Keynes

For centuries, many have tried to peer into the future. Soothsayers, oracles, palm readers, card readers, bone throwers, and even economists try to foretell what will be. You can join this elite group by using your computer and "Least-Squares Forecasting." Just type in the program and run it.

Let's try some elementary forecasting and make a prediction about the level of interest rates in 1986. To use this program, you have to estimate what one variable will be in order to predict another. We'll assume that the rate of inflation is the key. That is, if the rate of inflation rises, interest rates will climb. But interest rates will fall if inflation slows, if our assumption is correct.

The table presents historical observations on the two variables. The treasury bill rate is the basis for the interest rate in our example, and the consumer price index (CPI) determines the rate of inflation. This is the data you'll use to make your forecast.

Interest and Inflation Rates

Year	3-Month Treasury Bill Rate, %	Annual Change in the CPI, %
1968	5.3	4.2
1969	6.7	5.4
1970	6.4	5.9
1971	4.3	4.2
1972	4.1	3.3
1973	7.0	6.2
1974	7.9	11.0
1975	5.8	9.2
1976	5.0	5.7
1977	5.3	6.5
1978	7.2	7.6
1979	10.0	11.3
1980	11.6	13.5
1981	14.1	10.3
1982	10.7	6.2
1983	8.6	3.2

Looking into the Future

First, enter observations for the dependent variable Y. In our example, that's the rate of interest, or treasury bill rate, given in the table. (If you wanted to forecast the rate of inflation instead of interest rates, you could enter the former as the dependent variable.) Type them in and hit the RETURN key (without typing in a number) to signal that you're through. Do the same with the observations of the explanatory, or independent, variable, X, entering the rates of inflation (annual change in the CPI) from the table.

You'll have an opportunity to review and edit the figures. Least-Squares Forecasting will then estimate the regression equation:

$$\text{Predicted Y} = B_0 + B_1 * X = 3.286 + 0.593 * X$$

where the first value (3.286) is the constant's estimated value and the second (0.593) is the coefficient's estimated value.

R-squared, or the coefficient of determination, is the proportion of variation in the dependent variable explained by the regression equation. This statistic always ranges from 0 to 1, with a value close to 1 suggesting that X explains Y fairly well.

To forecast the value of interest rates in 1986, press any key to change the display, hit the Y key, and then enter the rate of inflation expected in that year, such as 7 percent. The computer responds with:

Predicted Y = 7.437

95% Confidence Interval

Lower Bound = 2.472

Upper Bound = 12.402

In other words, you're forecasting roughly 7-1/2 percent interest rates in 1986. The 95 percent confidence interval means that it's 95 percent sure that the true interest rate will be covered by the range between 2.5 percent and 12.4 percent. This band is computed under the strict assumption that the value of X, or the rate of inflation in 1986, is known with certainty. Keep in mind, however, that this is only an estimate, or prediction, based on the established pattern. Notice how wide the range is of 95 percent confidence. Although Y is predicted, it could very well be higher or lower. Chances are, however, that it will fall somewhere in this band.

You can return to the previous display and enter another

value for X to see the effect on Y. Try forecasting some of your own business's figures, such as sales or production. You may be surprised at how accurate your estimates are.

Nevertheless, remember that these are only predictions, and that they are not the result of sophisticated financial forecasting methods. And even sophisticated formulas have failed.

Least-Squares Forecasting

```

100 REM LEAST-SQUARES FORECASTING
110 REM INITIALIZE
120 GOSUB 240
130 REM ENTER DATA
140 GOSUB 610
150 REM EDIT DATA
160 GOSUB 950
170 REM COMPUTE
180 GOSUB 1360
190 REM DISPLAY RESULTS
200 GOSUB 2120
210 REM FORECAST
220 GOSUB 2430
230 END
240 REM INITIALIZE
250 REM TITLE
260 GOSUB 300
270 REM GREETING
280 GOSUB 360
290 RETURN
300 REM TITLE
310 PRINT CHR$(21): TEXT: HOME
320 VTAB 12: HTAB 14: PRINT "Least-Squares
330 VTAB 13: HTAB 15: PRINT "Forecasting "
;
340 FOR TIME = 1 TO 2000: NEXT
350 RETURN
360 REM GREETING
370 REM MAXIMUM NUMBER OF OBSERVATIONS
380 DATA 150
390 READ NX
400 DIM X(NX,2)
410 V$(1) = "Y":V$(2) = "X"
420 HOME: PRINT " This program estimates
    a simple linear";
430 PRINT "regression equation.
440 PRINT
450 PRINT " Future values of the dependen
    t

```

```

460 PRINT "variable (Y) are predicted, bas
    ed on the";
470 PRINT "value of X that you enter.
480 PRINT
490 PRINT " A 95% confidence interval is
    generated";
500 PRINT "for the forecast.
510 PRINT
520 PRINT " A total of ";NX;" observation
    s on your
530 PRINT "equation are allowed. Change 1
    ine 380"
540 PRINT "for a different value.
550 VTAB 23: HTAB 14: PRINT "Press any key
    ";
560 GET S$
570 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT
580 REM DIGITS AFTER DECIMAL
590 DA = 3
600 RETURN
610 REM ENTER DATA
620 REM ON Y
630 GOSUB 670
640 REM ON X
650 GOSUB 840
660 RETURN
670 REM Y
680 HOME
690 PRINT "Please enter observations on th
    e
700 PRINT "dependent variable, Y. Hit the
    RETURN
710 PRINT "key when you're through."; CHR$
    (7)
720 N = NX
730 FOR J = 1 TO NX
740 VTAB 5: HTAB 11: PRINT SPC( 20)
750 VTAB 5: HTAB 1: PRINT "Y( ";J; TAB( 7)
    ";")= ";
760 INPUT V$
770 IF V$ = "" THEN N = J - 1:J = NX
780 IF V$ < > "" THEN X(J,1) = VAL (V$)
790 NEXT J
800 REM DEGREES OF FREEDOM
810 V = N - 2
820 IF V < 1 THEN VTAB 22: HTAB 1: PRINT
    "You need at least three observations
    !"; CHR$ (7);: GOTO 720
830 RETURN

```

```

840 REM X
850 HOME
860 PRINT "Please enter data on the explan
    atory
870 PRINT "variable, X."; CHR$ (7)
880 FOR J = 1 TO N
890 VTAB 4: HTAB 11: PRINT SPC( 20)
900 VTAB 4: HTAB 1: PRINT "X( ";J; TAB( 7)
    ;")= ";
910 INPUT X$
920 X(J,2) = VAL (X$)
930 NEXT J
940 RETURN
950 REM CORRECT DATA
960 FOR I = 1 TO 2
970 FOR J = 1 TO N STEP 10
980 REM DISPLAY
990 GOSUB 1040
1000 REM CORRECT
1010 GOSUB 1170
1020 NEXT J,I
1030 RETURN
1040 REM DISPLAY DATA
1050 HOME
1060 PRINT L$
1070 A$ = CHR$ (32)
1080 IF J > 10 THEN A$ = " more "
1090 PRINT "These are";A$;"values of ";V$(
    I);";"
1100 PRINT L$: PRINT
1110 R = 0
1120 FOR L = J TO J + 9
1130 IF L < = N THEN R = R + 1: INVERSE :
    PRINT CHR$ (R + 64);: NORMAL : PRINT
    " No. ";L; TAB( 10)"= ";X(L,I)
1140 NEXT L
1150 VTAB 16: HTAB 1: PRINT L$
1160 RETURN
1170 REM CORRECT DATA
1180 VTAB 21: HTAB 1: PRINT SPC( 39)
1190 VTAB 23: HTAB 1: PRINT SPC( 39)
1200 VTAB 19: HTAB 1: PRINT "Corrections (
    Y/N) ? "; CHR$ (7);
1210 GET S$
1220 IF S$ = "N" OR S$ = "n" THEN 1350
1230 IF S$ < > "Y" AND S$ < > "y" THEN 1
    200
1240 VTAB 21: HTAB 1: PRINT "Which letter
    ? "; CHR$ (7);
1250 GET S$

```

```

1260 A = ASC (S$)
1270 IF A > 90 THEN A = A - 32
1280 Q = A - 64
1290 IF Q < 1 OR Q > R THEN 1240
1300 VTAB (A - 60): HTAB 1: FLASH : PRINT
    CHR$ (A); CHR$ (7);: NORMAL
1310 VTAB 23: HTAB 1: INPUT "What should t
    he value be ? ";S$
1320 X(J + Q - 1,I) = VAL (S$)
1330 VTAB (A - 60): HTAB 12: PRINT SPC( 2
    0);: HTAB 12: PRINT X(J + Q - 1,I)
1340 VTAB (A - 60): HTAB 1: INVERSE : PRINT
    CHR$ (A);: NORMAL : GOTO 1180
1350 RETURN
1360 REM COMPUTE
1370 HOME
1380 VTAB 12: HTAB 15: FLASH : PRINT "COMP
    UTING";: NORMAL
1390 REM KEY SUMS
1400 GOSUB 1480
1410 REM COEFFICIENT & CONSTANT
1420 GOSUB 1580
1430 REM ANOVA TERMS
1440 GOSUB 1620
1450 REM TERMS FOR FORECASTS
1460 GOSUB 1730
1470 RETURN
1480 REM KEY SUMS
1490 SX = 0:SY = 0:XQ = 0:YQ = 0:CP = 0
1500 FOR I = 1 TO N
1510 SX = SX + X(I,2)
1520 SY = SY + X(I,1)
1530 XQ = XQ + X(I,2) ^ 2
1540 YQ = YQ + X(I,1) ^ 2
1550 CP = CP + X(I,1) * X(I,2)
1560 NEXT
1570 RETURN
1580 REM COEFFICIENT & CONSTANT
1590 B = (N * CP - SX * SY) / (N * XQ - SX *
    SX)
1600 A = (SY - B * SX) / N
1610 RETURN
1620 REM ANOVA TERMS
1630 TSS = YQ - SY * SY / N
1640 RSS = B * (CP - SX * SY / N)
1650 ESS = TSS - RSS
1660 REM ERROR VARIANCE & STANDARD ERROR
    OF THE ESTIMATE
1670 EV = ESS / V
1680 SEE = SQR (EV)

```

```

1690 REM STANDARD ERRORS OF THE COEFFICIE
    NT & CONSTANT
1700 SB = SQR (EV / (XQ - SX * SX / N))
1710 SA = SQR (EV * XQ / (N * XQ - SX * SX
    ))
1720 RETURN
1730 REM FORECAST TERMS
1740 REM MEAN OF X
1750 GOSUB 1790
1760 REM t-VALUES
1770 GOSUB 1890
1780 RETURN
1790 REM MEAN OF X
1800 REM MEAN
1810 SUM = 0
1820 FOR I = 1 TO N
1830 SUM = SUM + X(I,2)
1840 NEXT I
1850 XBAR = SUM / N
1860 REM SUM OF SQUARED DEVIATIONS
1870 SD = XQ - SX * SX / N
1880 RETURN
1890 REM t-VALUES
1900 REM VALUES FOR APPROXIMATION FORMULA
1910 DATA 1.96,0.60033,0.9591,-0.90259,0.
    11588
1920 READ A1,B1,C1,D1,E1
1930 REM COMPUTE -- ACTUAL VALUES FOR LES
    S THAN 4 DEGREES OF FREEDOM
1940 IF V < 4 THEN GOSUB 1970
1950 IF V > = 4 THEN GOSUB 2040
1960 RETURN
1970 REM ACTUAL VALUES
1980 DATA 12.706,4.303,3.182
1990 FOR I = 1 TO 3
2000 READ TV
2010 IF V = I THEN T = TV
2020 NEXT I
2030 RETURN
2040 REM APPROXIMATION
2050 REM NUMERATOR
2060 T1 = A1 * V + B1 + C1 / V
2070 REM DENOMINATOR
2080 T2 = V + D1 + E1 / V
2090 REM QUOTIENT
2100 T = T1 / T2
2110 RETURN
2120 REM DISPLAY RESULTS
2130 REM EQUATION
2140 GOSUB 2180

```

```

2150 REM ANOVA TERMS
2160 GOSUB 2320
2170 RETURN
2180 REM EQUATION
2190 HOME
2200 PRINT L$
2210 PRINT TAB( 11)"REGRESSION RESULTS
2220 PRINT L$
2230 VTAB 5: HTAB 15: PRINT "Estimated"; TAB(
32)"Standard
2240 VTAB 6: HTAB 1: PRINT "Term"; TAB( 17
)"Value"; TAB( 34)"Error
2250 VTAB 8: HTAB 1: PRINT "B0"
2260 VTAB 8:HT = 23:NR = A: GOSUB 9000
2270 VTAB 8:HT = 39:NR = SA: GOSUB 9000
2280 VTAB 9: HTAB 1: PRINT "B1"
2290 VTAB 9:HT = 23:NR = B: GOSUB 9000
2300 VTAB 9:HT = 39:NR = SB: GOSUB 9000
2310 RETURN
2320 REM ANOVA TERMS
2330 VTAB 11: HTAB 1: PRINT "Number of
2340 VTAB 12: HTAB 1: PRINT "Observations"
; TAB( 17)"=";:HT = 32:NR = N: GOSUB 9
000
2350 VTAB 14: HTAB 1: PRINT "R-Squared"; TAB(
17)"=";:NR = RSS / TSS: GOSUB 9000
2360 VTAB 16: HTAB 1: PRINT "Standard Erro
r
2370 VTAB 17: HTAB 1: PRINT "of the Estima
te"; TAB( 17)"=";:NR = SEE: GOSUB 9000

2380 VTAB 19: HTAB 1: PRINT "F-Statistic";
TAB( 17)"=";:NR = RSS / EV: GOSUB 900
0
2390 VTAB 21: HTAB 1: PRINT L$
2400 VTAB 23: HTAB 14: PRINT "Press any ke
y ";
2410 GET S$
2420 RETURN
2430 REM FORECASTS
2440 REM HEADING
2450 GOSUB 2510
2460 REM PREDICTION
2470 GOSUB 2570
2480 REM CONFIDENCE INTERVAL
2490 IF S$ = "Y" OR S$ = "y" THEN GOSUB 2
680: GOTO 2450
2500 RETURN
2510 REM HEADING
2520 HOME

```

```

2530 PRINT L$
2540 PRINT TAB( 15)"FORECASTS
2550 PRINT L$
2560 RETURN
2570 REM PREDICTION
2580 VTAB 6: HTAB 1: PRINT "Would you like
      to forecast the value of
2590 VTAB 7: HTAB 1: PRINT "the dependent
      variable (Y/N) ? "; CHR$( 7);
2600 GET S$
2610 IF S$ = "N" OR S$ = "n" THEN 2670
2620 IF S$ < > "Y" AND S$ < > "y" THEN 2
      590
2630 VTAB 9: HTAB 1: INPUT "What is the va
      lue of X ? ";X$
2640 X = VAL (X$)
2650 PD = A + B * X
2660 VTAB 12: HTAB 1: INVERSE : PRINT "PRE
      DICTED Y";: NORMAL : HTAB 21: PRINT "="
      ;:HT = 36:NR = PD: GOSUB 9000
2670 RETURN
2680 REM CONFIDENCE INTERVAL
2690 VTAB 14: HTAB 1: INVERSE : PRINT "95%
      CONFIDENCE BAND": NORMAL
2700 REM FORECAST VARIANCE
2710 FV = EV * (1 + 1 / N + (X - XBAR) ^ 2 /
      SD)
2720 VTAB 15: HTAB 5: PRINT "Lower Bound";
      TAB( 21)"=";:NR = PD - T * SQR (FV):
      GOSUB 9000
2730 VTAB 16: HTAB 5: PRINT "Upper Bound";
      TAB( 21)"=";:NR = PD + T * SQR (FV):
      GOSUB 9000
2740 VTAB 22: HTAB 1: PRINT L$
2750 VTAB 23: HTAB 14: PRINT "Press any ke
      y ";
2760 GET S$
2770 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
      00000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
      1)
9060 LET ZL = INT (ZN):ZR = INT ((ZN - Z
      L) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
      Z$ + STR$ (ZR),DA)

```



```
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
    "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
    ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN
```

Time-Series Forecasting

If you want to be able to forecast future values using one of several different estimating procedures, "Time-Series Forecasting" will be useful. It's a flexible and powerful tool any business person will find valuable.

The problem with least-squares forecasting is that future values of one variable are needed before future values of another can be predicted. In the preceding example, for instance, you needed to know the future rate of inflation to forecast what interest rates will be.

Time-series forecasting skirts this problem by making *future predictions* of a variable using *past observations* of that same term. A time series, incidentally, is a set of observations on a variable, in chronological order, at a set frequency. Monthly rainfall, yearly income, and daily stock market prices are all examples.

"Time-Series Forecasting" lets you use any of these methods to estimate the future value of a variable:

- Least-squares trend
- Semiaverages
- Percent change
- First difference
- Past averages

You can forecast as far into the future as you dare to go, using one technique after another. Let's explain each briefly.

Multiple Predictions

Type in and run the program. You'll need to enter some data first of all. If you don't have figures of your own to use, try the following example.

We want to predict the value of the gross national product (GNP) through 1985. The previous 13 years' figures are shown in the table.

Gross National Product (GNP)

Year	GNP (Billions)
1972	\$1,186
1973	1,326
1974	1,434
1975	1,549
1976	1,718
1977	1,918
1978	2,164
1979	2,418
1980	2,633
1981	2,938
1982	3,069
1983	3,305
1984*	3,645

* Preliminary

You need to enter data for only one variable with this program. Type in the figures on the right of the table (omitting the commas and dollar signs). Signal the computer that you're finished by hitting the RETURN key without typing in a number. You'll be able to check and change the figures; then the display changes to show you the available forecasting options.

Least-squares trend. The program forecasts by extending a regression line into the future, fitting a least-squares line between the GNP and time. Observations on time are generated by the computer, with 1972 corresponding to 1, 1973 corresponding to 2, and so on. Since you've entered 13 time periods, to estimate the GNP for 1985 and 1986, you need to request 2 future time periods. The computer shows the estimated GNP for time periods 14 and 15 (1985 and 1986) as approximately \$3,704 and \$3,911 billion, respectively.

Semiaverages. The time series is divided into two roughly equal parts, with means computed for each. A straight line is then fitted through the two points, with values on the line representing forecasts. Using this method, the GNP in 1985 and 1986 is estimated as \$3,719 and \$3,928 billion, respectively.

Percent change. This method works as its name implies. Specifically, the computer calculates the change between the

last two values in the time series and then applies the percentage to future years. Hence, the predicted GNP in 1985 is \$4,020 billion. That figure is used to predict the 1986 GNP: \$4,434 billion.

First difference. This works a lot like the percent change method. The difference (instead of percent change) between the last two values of the time series is used as a factor to forecast future values. Hence, GNP forecasts for 1985 and 1986 are \$3,985 billion and \$4,325 billion, respectively.

Past averages. Percent changes and first differences in this method are computed using any number of past calculations, not just one. You can select between three types of calculations, and even specify how many time periods you'll use as a base. For example, the mean of the past two first differences is $(\$3,305 - \$3,069) + (\$3,985 - \$3,305)$, all divided by 2, giving a total of \$288. The predicted GNP for 1985 is then $\$3,645 + \$288 = \$3,933$ billion.

Clearly, every time-series forecast depends on the adage "The past is prologue." If that cliché is true, you'll be able to use this program to your advantage, perhaps helping you make business decisions with more confidence.

Time-Series Forecasting

```

100 REM TIME-SERIES FORECASTING
110 REM INITIALIZE
120 GOSUB 240
130 REM ENTER DATA
140 GOSUB 620
150 REM EDIT DATA
160 GOSUB 770
170 REM CHOOSE METHOD
180 GOSUB 1150
190 IF CHOICE = 6 THEN 230
200 REM FORECAST
210 GOSUB 1270
220 GOTO 180
230 END
240 REM INITIALIZE
250 REM TITLE
260 GOSUB 320
270 REM GREETING
280 GOSUB 410
290 REM CHOICES
300 GOSUB 580
310 RETURN
320 REM TITLE
    
```

```

330 PRINT CHR$(21): TEXT : HOME
340 VTAB 12: HTAB 15: PRINT "Time-Series
350 VTAB 13: HTAB 16: PRINT "Forecasting "
    ;
360 FOR TIME = 1 TO 1500: NEXT
370 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT
380 REM DIGITS AFTER DECIMAL
390 DA = 3
400 RETURN
410 REM GREETING
420 REM MAX NUMBER OF OBSERVATIONS
430 DATA 150
440 READ NX
450 DIM Y(NX)
460 HOME
470 PRINT "This program forecasts future v
    alues of
480 PRINT "a time series using a host of t
    rend-
490 PRINT "analysis techniques.
500 PRINT
510 PRINT "The maximum number of observati
    ons
520 PRINT "allowed is ";NX;".
530 PRINT
540 PRINT "Change line 430 for a different
    value.
550 VTAB 23: HTAB 14: PRINT "Press any key
    ";
560 GET S$
570 RETURN
580 REM CHOICES
590 DATA Least-Squares Trend, Semi-Avera
    ges, Percent Changes, First Difference
    s, Past Averages, None
600 FOR I = 1 TO 6: READ C$(I): NEXT
610 RETURN
620 REM ENTER DATA
630 HOME
640 PRINT "Please enter observations on yo
    ur time
650 PRINT "series. Hit the RETURN key whe
    n you're
660 PRINT "through.
670 N = NX
680 FOR J = 1 TO NX
690 VTAB 5: HTAB 11: PRINT SPC(20)
700 VTAB 5: HTAB 1: PRINT "T( ";J; TAB(7)
    ;")= ";

```

```

710 INPUT V$
720 IF V$ = "" THEN N = J - 1:J = NX
730 IF V$ < > "" THEN Y(J) = VAL (V$)
740 NEXT J
750 IF N < 3 THEN VTAB 22: HTAB 1: PRINT
    "You need at least three observations
    !"; CHR$ (7);: GOTO 670
760 RETURN
770 REM CORRECT DATA
780 FOR J = 1 TO N STEP 10
790 REM DISPLAY
800 GOSUB 850
810 REM CORRECT
820 GOSUB 960
830 NEXT J
840 RETURN
850 REM DISPLAY DATA
860 HOME
870 PRINT L$
880 PRINT "These are values of your time s
    eries.
890 PRINT L$: PRINT
900 R = 0
910 FOR L = J TO J + 9
920 IF L < = N THEN R = R + 1: INVERSE : PRINT
    CHR$ (R + 64);: NORMAL : PRINT " No.
    ";L; TAB( 10)"=";Y(L)
930 NEXT L
940 VTAB 16: HTAB 1: PRINT L$
950 RETURN
960 REM CORRECT DATA
970 VTAB 21: HTAB 1: PRINT SPC( 39)
980 VTAB 23: HTAB 1: PRINT SPC( 39)
990 VTAB 19: HTAB 1: PRINT "Corrections (Y
    /N) ? "; CHR$ (7);
1000 GET S$
1010 IF S$ = "N" OR S$ = "n" THEN 1140
1020 IF S$ < > "Y" AND S$ < > "y" THEN 9
    90
1030 VTAB 21: HTAB 1: PRINT "Which letter
    ? "; CHR$ (7);
1040 GET S$
1050 A = ASC (S$)
1060 IF A > 90 THEN A = A - 32
1070 Q = A - 64
1080 IF Q < 1 OR Q > R THEN 1030
1090 VTAB (A - 60): HTAB 1: FLASH : PRINT
    CHR$ (A); CHR$ (7);: NORMAL
1100 VTAB 23: HTAB 1: INPUT "What should t
    he value be ? ";S$

```

```

1110 Y(J + Q - 1) = VAL (S$)
1120 VTAB (A - 60): HTAB 12: PRINT SPC( 2
    0);: HTAB 12: PRINT Y(J + Q - 1)
1130 VTAB (A - 60): HTAB 1: INVERSE : PRINT
    CHR$ (A);: NORMAL : GOTO 970
1140 RETURN
1150 REM CHOOSE METHOD
1160 HOME
1170 PRINT "Future values of your time ser
    ies are
1180 PRINT "projected using any of these m
    ethods:
1190 FOR I = 1 TO 6
1200 VTAB 2 * I + 2: HTAB 8: INVERSE : PRINT
    I;: NORMAL : HTAB 11: PRINT C$(I)
1210 NEXT
1220 VTAB 18: HTAB 8: PRINT "Your Choice =
    ? "; CHR$ (7);
1230 GET S$
1240 CHOICE = VAL (S$)
1250 IF CHOICE < 1 OR CHOICE > 6 THEN 1220
1260 RETURN
1270 REM FORECAST
1280 REM NUMBER OF FUTURE PERIODS
1290 GOSUB 1370
1300 REM PROJECTIONS
1310 HOME
1320 IF CHOICE < > 5 THEN VTAB 12: HTAB
    15: FLASH : PRINT "FORECASTING": NORMAL
    : FOR PAUSE = 1 TO 1500: NEXT
1330 ON CHOICE GOSUB 1530,1710,1980,2030,2
    080
1340 REM DISPLAY
1350 GOSUB 2510
1360 RETURN
1370 REM NUMBER OF FUTURE PERIODS
1380 HOME
1390 PRINT "The last period of your time s
    eries is
1400 PRINT "number ";N;". "
1410 PRINT
1420 PRINT "Please enter the number of fut
    ure
1430 PRINT "periods to forecast.
1440 VTAB 7: HTAB 12: PRINT SPC( 20); CHR$
    (7)
1450 VTAB 7: HTAB 1: INPUT "Number = ? ";N
    $
1460 NF = INT ( VAL (N$))
1470 VTAB 21: HTAB 1: PRINT SPC( 40)

```

```

1480 IF NF < 1 THEN VTAB 21: HTAB 2: PRINT
      "It's too easy to forecast the past !"
      : GOTO 1440
1490 REM CHECK FOR MEMORY CONSTRAINT
1500 T = N + NF
1510 IF T > NX THEN VTAB 21: HTAB 2: PRINT
      "Only ";NX - N;" more periods are allo
      wed !": GOTO 1440
1520 RETURN
1530 REM OLS
1540 REM KEY SUMS
1550 SX = 0:SY = 0:XQ = 0:YQ = 0:XY = 0
1560 FOR I = 1 TO N
1570 SX = SX + I
1580 SY = SY + Y(I)
1590 XQ = XQ + I * I
1600 YQ = YQ + Y(I) * Y(I)
1610 XY = XY + Y(I) * I
1620 NEXT
1630 REM ESTIMATES
1640 B = (N * XY - SX * SY) / (N * XQ - SX *
      SX)
1650 A = (SY - B * SX) / N
1660 REM FORECASTS
1670 FOR I = N + 1 TO T
1680 Y(I) = A + B * I
1690 NEXT
1700 RETURN
1710 REM SEMI-AVERAGES
1720 REM NUMBER OF POINTS IN EACH GROUP
1730 G1 = INT (N / 2)
1740 G2 = N - G1
1750 REM GROUP MEANS
1760 REM FIRST
1770 Y1 = 0:X1 = 0
1780 FOR I = 1 TO G1
1790 Y1 = Y1 + Y(I)
1800 X1 = X1 + I
1810 NEXT
1820 Y1 = Y1 / G1:X1 = X1 / G1
1830 REM SECOND
1840 Y2 = 0:X2 = 0
1850 FOR I = G1 + 1 TO N
1860 Y2 = Y2 + Y(I)
1870 X2 = X2 + I
1880 NEXT
1890 Y2 = Y2 / G2:X2 = X2 / G2
1900 REM ESTIMATES
1910 B = (Y2 - Y1) / (X2 - X1)
1920 A = Y2 - B * X2

```



```

1930 REM FORECASTS
1940 FOR I = N + 1 TO T
1950 Y(I) = A + B * I
1960 NEXT
1970 RETURN
1980 REM PERCENT CHANGE
1990 FOR I = N + 1 TO T
2000 Y(I) = Y(I - 1) * Y(I - 1) / Y(I - 2)
2010 NEXT
2020 RETURN
2030 REM FIRST DIFFERENCES
2040 FOR I = N + 1 TO T
2050 Y(I) = Y(I - 1) + (Y(I - 1) - Y(I - 2)
)
2060 NEXT
2070 RETURN
2080 REM PAST AVERAGE
2090 T$(1) = "Percent Changes":T$(2) = "First
Differences":T$(3) = "Actual Values
"
2100 PRINT "Would you like to use an average
of past
2110 FOR I = 1 TO 3
2120 VTAB I * 2 + 2: HTAB 10: INVERSE : PRINT
I;: NORMAL : HTAB 12: PRINT T$(I)
2130 NEXT
2140 VTAB 11: HTAB 10: PRINT "Choice = ? "
; CHR$(7);
2150 GET S$
2160 PICK = VAL (S$)
2170 IF PICK < 1 OR PICK > 3 THEN 2140
2180 REM SELECT NUMBER OF PAST VALUES
2190 GOSUB 2260
2200 REM FORECAST
2210 HOME
2220 VTAB 12: HTAB 15: FLASH : PRINT "FORE
CASTING": NORMAL
2230 FOR PAUSE = 1 TO 1500: NEXT
2240 GOSUB 2350
2250 RETURN
2260 REM NUMBER OF PAST VALUES
2270 VTAB 15: HTAB 1: PRINT "How many past
";T$(PICK);" do
2280 VTAB 16: HTAB 19: PRINT SPC(15); CHR$(
7)
2290 VTAB 16: HTAB 1: INPUT "you want to use
? ";P$
2300 PP = INT ( VAL (P$))
2310 IF PP < 1 THEN 2280
2320 IF PICK < 3 AND PP + 1 > N THEN VTAB

```

```

23: HTAB 8: PRINT "Only ";N - 1;" are
available !": GOTO 2280
2330 IF PICK = 3 AND PP > N THEN VTAB 23:
HTAB 8: PRINT "Only ";N;" are availab
le !": GOTO 2280
2340 RETURN
2350 REM FORECASTS
2360 REM PAST AVERAGE
2370 AVG = 0
2380 FOR I = N TO N - PP + 1 STEP - 1
2390 IF PICK = 1 THEN AVG = AVG + Y(I) / Y
(I - 1)
2400 IF PICK = 2 THEN AVG = AVG + Y(I) - Y
(I - 1)
2410 IF PICK = 3 THEN AVG = AVG + Y(I)
2420 NEXT
2430 AVG = AVG / PP
2440 REM FORECASTS
2450 FOR I = N + 1 TO T
2460 IF PICK = 1 THEN Y(I) = Y(I - 1) * AV
G
2470 IF PICK = 2 THEN Y(I) = Y(I - 1) + AV
G
2480 IF PICK = 3 THEN Y(I) = AVG
2490 NEXT
2500 RETURN
2510 REM DISPLAY
2520 FOR L = 0 TO INT ((NF - 1) / 10)
2530 REM HEADING
2540 GOSUB 2590
2550 REM BODY
2560 GOSUB 2670
2570 NEXT L
2580 RETURN
2590 REM HEADING
2600 HOME
2610 PRINT L$
2620 PRINT TAB( 15)"FORECASTS
2630 PRINT L$
2640 VTAB 4: HTAB 1: PRINT "Method: ";C$(C
HOICE);" ";
2650 IF CHOICE = 5 THEN PRINT "of ";PP: VTAB
5: HTAB 9: PRINT T$(PICK)
2660 RETURN
2670 REM BODY
2680 FOR J = 1 TO 10
2690 M = J + L * 10 + N
2700 IF M < = T THEN VTAB J + 8: HTAB 1:
PRINT "Time Period ( ";M; TAB( 18)" )=
";:HT = 34:NR = Y(M): GOSUB 9000

```

```

2710 NEXT J
2720 VTAB 22: HTAB 1: PRINT L$
2730 VTAB 23: HTAB 14: PRINT "Press any key";
2740 GET S$
2750 RETURN
9000 REM "PRINT USING" SUBROUTINE
9010 ZR$ = "": ZS$ = "": ZD$ = "": ZZ$ = "00000000"
9020 N$ = STR$ (NR)
9030 IF DOLL$ = "YES" THEN ZD$ = "$"
9040 IF NR < 0 THEN ZS$ = "-"
9050 LET ZN = ABS (NR) + 5 * 10 ^ - (DA + 1)
9060 LET ZL = INT (ZN): ZR = INT ((ZN - ZL) * 10 ^ DA)
9070 IF DA > 0 THEN ZR$ = "." + RIGHT$ (ZL + STR$ (ZR), DA)
9080 ZT$ = " " + STR$ (ZL)
9090 IF MID$ (ZT$, LEN (ZT$) - 3, 1) < "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) + ZR$
9100 HTAB (HT + 1 - LEN (N$)): PRINT N$
9110 RETURN

```

Computer Cash Register

Turn your computer into a fast-working and easy-to-use cash register with this program. It automatically figures sales tax and even tallies the receipts at the end of the day.

If you have a small business, and you're tired of looking at sales-tax tables when you ring up purchases, you'll find this program a joy to use. All you have to do is enter the price of the item and the computer responds with the total payment due, including tax. After you type in the amount of money received from the customer, the program tells you how much change to give. It's as easy as that.

After you've run the program, enter the sales tax of your area. You can use decimal fractions, such as 4.5 percent, if necessary. The cash register screen display appears and you can begin. Type in the price of the item and then press the RETURN key. A total of this transaction, tax included, is shown at the top of the screen. Now, enter the amount given to you by the customer, and the change you need to return displays.

If you make a mistake entering the transaction, just press R and you have another chance. No more scratching out entries on the cash register slip. You can even view total sales through any point in time by pressing T.

Computer Cash Register

```
100 REM  COMPUTER CASH REGISTER
110 REM  INITIALIZE
120 GOSUB 180
130 REM  DRAW BOX
140 GOSUB 360
150 REM  OPERATE
160 GOSUB 680
170 END
180 REM  INITIALIZE
190 PRINT CHR$(21): TEXT : HOME
200 PRINT "  This program turns your compu
    ter into
210 PRINT "a cash register.  Please enter
    the
220 PRINT "percentage sales tax in your ar
    ea.
```

```

230 PRINT
240 PRINT " For example, enter 5 for 5%,
    7 for
250 PRINT "7%, and so on.
260 VTAB 8: HTAB 15: PRINT SPC( 15); CHR$
    (7)
270 VTAB 8: HTAB 1: INPUT "Sales Tax = ? "
    ;T$
280 T = VAL (T$)
290 IF T < 0 THEN VTAB 23: HTAB 13: PRINT
    "No such luck !": GOTO 260
300 TR = 1 + T / 100
310 SALES = 0:N = 0:TAXES = 0
320 L$ = "": FOR I = 1 TO 39:L$ = L$ + "=":
    NEXT
330 REM DIGITS AFTER DECIMAL
340 DA = 2
350 RETURN
360 REM DRAW BOX
370 REM BOX
380 GOSUB 420
390 REM LABELS
400 GOSUB 570
410 RETURN
420 REM BOX
430 HOME : INVERSE
440 FOR I = 1 TO 39
450 VTAB 1: HTAB I: PRINT CHR$ (32)
460 VTAB 2: HTAB I: PRINT CHR$ (32)
470 VTAB 3: HTAB I: PRINT CHR$ (32)
480 VTAB 16: HTAB I: PRINT CHR$ (32)
490 VTAB 24: HTAB I: PRINT CHR$ (32);
500 NEXT
510 FOR ROW = 1 TO 23
520 VTAB ROW: HTAB 1: PRINT CHR$ (32)
530 VTAB ROW: HTAB 39: PRINT CHR$ (32)
540 NEXT ROW
550 NORMAL
560 RETURN
570 REM LABELS
580 VTAB 2: HTAB 6: PRINT "$$ COMPUTER CAS
    H REGISTER $$"
590 VTAB 18: HTAB 6: PRINT "PRESS"; TAB( 2
    7)"TO
600 VTAB 20: HTAB 8: PRINT "C"; TAB( 22)"C
    ontinue
610 VTAB 21: HTAB 8: PRINT "R"; TAB( 22)"R
    edo last entry
620 VTAB 22: HTAB 8: PRINT "T"; TAB( 22)"T
    ally totals
    
```

```

630 VTAB 7: HTAB 10: PRINT "Price = $
640 VTAB 8: HTAB 7: PRINT "With Tax = $
650 VTAB 11: HTAB 8: PRINT "Payment = $
660 VTAB 13: HTAB 9: PRINT "Change = $
670 RETURN
680 REM OPERATE
690 REM ENTER PRICE & PAYMENT
700 GOSUB 850
710 REM ENTER NEXT ACTION
720 GOSUB 1110
730 REM REDO
740 IF R$ = "R" OR R$ = "r" THEN VTAB 5: HTAB
3: PRINT "Let's try that again !"; CHR$
(7): GOTO 700
750 VTAB 5: HTAB 3: PRINT SPC( 22)
760 REM ADD TO TOTALS
770 GOSUB 1200
780 REM CONTINUE
790 IF R$ = "C" OR R$ = "c" THEN GOTO 700
800 REM DISPLAY TOTALS
810 IF R$ = "T" OR R$ = "t" THEN GOSUB 12
50
820 REM CONTINUE OPERATING
830 IF S$ = "Y" OR S$ = "y" THEN GOSUB 36
0: GOTO 700
840 RETURN
850 REM ENTER FIGURES
860 REM BLANK-OUT AREAS
870 VTAB 7: HTAB 19: PRINT SPC( 20)
880 VTAB 8: HTAB 19: PRINT SPC( 20)
890 VTAB 11: HTAB 19: PRINT SPC( 20)
900 VTAB 13: HTAB 19: PRINT SPC( 20)
910 REM ENTER PRICE
920 VTAB 7: HTAB 20: PRINT CHR$ (7);: INPUT
" ";D$
930 VTAB 7: HTAB 39: INVERSE : PRINT CHR$
(32): NORMAL
940 PRICE = VAL (D$)
950 IF PRICE < = 0 THEN VTAB 7: HTAB 21:
PRINT "No free lunch !";: FOR TIME =
1 TO 250:P = PEEK ( - 16336): NEXT TI
ME: GOTO 870
960 PWT = INT ((PRICE * TR + .005) * 100) /
100
970 VTAB 8: HTAB 21: PRINT PWT
980 REM PAYMENT
990 VTAB 11: HTAB 20: PRINT CHR$ (7);: INPUT
" ";D$
1000 VTAB 11: HTAB 39: INVERSE : PRINT CHR$
(32): NORMAL

```

```

1010 PAYMENT = VAL (D$)
1020 IF PAYMENT > = PWT THEN 1080
1030 VTAB 11: HTAB 21: PRINT "I want more
!";
1040 FOR TIME = 1 TO 250
1050 P = PEEK ( - 16336)
1060 NEXT TIME
1070 VTAB 11: HTAB 19: PRINT SPC( 20): GOTO
990
1080 CHANGE = INT ((PAYMENT - PWT + .005) *
100) / 100
1090 VTAB 13: HTAB 21: PRINT CHANGE
1100 RETURN
1110 REM ENTER NEXT ACTION
1120 VTAB 18: HTAB 6: FLASH : PRINT "PRESS
"; CHR$ (7);: NORMAL
1130 P = PEEK ( - 16384)
1140 IF P < 128 THEN 1130
1150 R$ = CHR$ (P - 128)
1160 POKE - 16368,0
1170 IF R$ < > "C" AND R$ < > "c" AND R$
< > "R" AND R$ < > "r" AND R$ < >
"T" AND R$ < > "t" THEN 1120
1180 VTAB 18: HTAB 6: PRINT "PRESS"
1190 RETURN
1200 REM ADD TO TOTALS
1210 SALES = SALES + PRICE
1220 TAXES = TAXES + (PWT - PRICE)
1230 N = N + 1
1240 RETURN
1250 REM DISPLAY TOTALS
1260 HOME
1270 PRINT L$
1280 PRINT TAB( 17)"TOTALS
1290 PRINT L$
1300 VTAB 5: HTAB 1: PRINT "Total Sales"; TAB(
17);"=";:DOLL$ = "YES":HT = 32:NR = SA
LES: GOSUB 9000
1310 VTAB 7: HTAB 1: PRINT "Total Taxes"; TAB(
17);"=";:NR = TAXES: GOSUB 9000
1320 VTAB 9: HTAB 1: PRINT "Sum"; TAB( 17)
;"=";:NR = SALES + TAXES: GOSUB 9000
1330 VTAB 14: HTAB 1: PRINT "Number of Sal
es"; TAB( 17)"=";:DOLL$ = "NO":NR = N:
GOSUB 9000
1340 VTAB 20: HTAB 1: PRINT L$
1350 VTAB 22: HTAB 6: PRINT "Keep cash reg
ister on (Y/N) ? ";
1360 GET S$
1370 IF S$ < > "Y" AND S$ < > "v" AND S$

```

```

        < > "N" AND S$ < > "n" THEN 1360
1380  RETURN
9000  REM "PRINT USING" SUBROUTINE
9010  ZR$ = "":ZS$ = "":ZD$ = "":ZZ$ = "0000
      00000"
9020  N$ = STR$ (NR)
9030  IF DOLL$ = "YES" THEN ZD$ = "$"
9040  IF NR < 0 THEN ZS$ = "-"
9050  LET ZN = ABS (NR) + 5 * 10 ^ - (DA +
      1)
9060  LET ZL = INT (ZN):ZR = INT ((ZN - Z
      L) * 10 ^ DA)
9070  IF DA > 0 THEN ZR$ = "." + RIGHT$ (Z
      Z$ + STR$ (ZR),DA)
9080  ZT$ = " " + STR$ (ZL)
9090  IF MID$ (ZT$, LEN (ZT$) - 3,1) < >
      "E" THEN N$ = ZS$ + ZD$ + STR$ (ZL) +
      ZR$
9100  HTAB (HT + 1 - LEN (N$)): PRINT N$
9110  RETURN

```


Beginner's Guide to Typing In Programs

What Is a Program?

A computer cannot perform any task by itself. Like a car without gas, a computer has *potential*, but without a program, it isn't going anywhere. Most of the programs published in this book are written in a computer language called Applesoft BASIC. It is easy to learn and works on the Apple II, II+, IIe, and IIfx.

BASIC Programs

Computers can be picky. Unlike the English language, which is full of ambiguities, BASIC usually has only one right way of stating something. Every letter, character, or number is significant. Common mistakes are substituting the letter O for the numeral 0, a lowercase l for the numeral 1, or an uppercase B for the numeral 8. Also, be sure to enter all punctuation marks, such as colons and commas, just as they appear in the book. Spacing can be important. To be safe, type in the listings *exactly* as they appear.

About DATA Statements

Some programs contain a section, or sections, of DATA statements. These lines provide information needed by the program. Some DATA statements contain actual programs (called machine language); others contain graphics codes. These lines are especially sensitive to errors.

If a single number in any one DATA statement is mistyped, your machine could lock up, or crash. The keyboard may seem dead and the screen may go blank. But don't panic; no damage is done. To regain control, you have to reset your computer (in effect, you turn it off and then turn it back on). That will erase whatever program was in memory, *so always save a copy of your program before you run it*. If your computer crashes, you can load the program and look for your mistake.

Sometimes a mistyped DATA statement will cause an error message when the program is run. The error message may refer to the program line that READs the data. *The error is still in the DATA statements, though.*

Get to Know Your Machine

You should familiarize yourself with your computer before attempting to type in a program. Learn the statements you use to store and retrieve programs from tape or disk. You'll want to save a copy of your program so that you won't have to type it in every time you want to use it. Learn to use your machine's editing functions. How do you change a line if you made a mistake? You can always retype the line, but you at least need to know how to backspace. Do you know how to enter special characters? It's all explained in your computer's manuals.

A Quick Review

1. Type in the program a line at a time in order. Press RETURN at the end of each line. Use backspace or the back-arrow to correct mistakes.
2. Check the line you've typed against the line in the book. You can check the entire program again if you get an error when you run the program.

Index

- accuracy of fit statistics 301-2
- analysis of variance 284-85
 - illustration of 285
- "Analysis of Variance" program 284-89
- bar chart, computation of 276-83
 - illustration of 276
- "Beans" program 90-98
- "Calorie Cop" program 170-77
- "Car Payments" program 109-12
- "Car Trek" program 151-58
- "Chomp" program 13-21
- coefficient 290-301
- "College Planner" program 120-28
- column vector 219. *See also* matrices, dimensions
- "Computer Cash Register" program 382-86
- constant 300-301
- correlation coefficient 290-93. *See also* rho
- cumulative frequency 262-64
- database files 337-39
- "Database Management" program 337-48
- degrees of freedom 285
- dependent variable 302, 310, 363
- derivative 214-15
- "Derivative" program 214-18
- Durbin-Watson statistic 312
- effective interest rates 129-30
- "Effective Yield on Investments" program 129-32
- elements 219. *See also* matrices
- equations
 - exponential 322, 324-25
 - illustrations of 297
 - linear 322, 324-25
 - power 322, 324-25
 - reciprocal 322, 324-25
 - regression 324, 363
- explanatory variable 302, 309, 363
- frequency 263-64, 276-77
- "Frequency Plot" program 276-83
- F-statistics 285, 302, 311
- function 214
- future worth, of a dollar 349
 - illustration of 349
- "General-Form Curve Fitter" program 322-32
- gold and silver prices 291-92
- Gross National Product (GNP) 372-74
- "Guess a Word" program 5-12
- "Hi-Q" program 80-89
- "Household Budget" program 142-50
- identity matrix 235, 236. *See also* matrices, inversion of
- independent variable 302
- integral 206
- integration 206-11
- internal rate of return (IRR) 355
- "Internal Rate of Return" program 354-61
- inverse 235-36
 - mechanics of inversion, illustration of 236
 - See also* matrices, inversion of
- inverse relationship of variables 299
- investments 103-5, 129-30, 133-35, 362-64
 - value of 349-50, 354-56
- IRA (Individual Retirement Account) 103-5
- "IRA Planner" program 103-8
- lag matrix 227. *See also* matrices, multiplication
- lead matrix 227. *See also* matrices, multiplication
- least squares forecasting 362-64, 372
- "Least Squares Forecasting" program 362-71
- linear association 290, 292, 302, 312, 322
- "Matches" program 31-38
- "Math Man" program 22-30
- "MatMan" program 242-51
- matrices 219-44
 - addition 220-21, 242-43
 - dimensions 219-20
 - illustrations of 219, 220, 221, 228, 229, 235, 242
 - inversion 235-37, 242-43
 - multiplication 227-29, 242-43
 - subtraction 220-21, 242-43
- "Matrix Addition and Subtraction" program 219-26
- "Matrix Inversion" program 235-41
- "Matrix Multiplication" program 227-34
- mean 202, 257-58
- "Mean, Variance, and Standard Deviation" program 257-61
- median 202
- mode 202
- "Mortgage Payments" program 113-19
- "Mosaic Puzzle" program 61-70
- multiple linear regression 308-12
- "Multiple Linear Regression Analysis" program 308-21
- municipal bond 133
- "Municipal Bond Buyer" program 133-41

- "Net Present Value" program 349-53
- nominal interest rates 129-30
- normal curve 202
 - illustration of 202
- normal distribution 201-2
- "Numerical Integration" program 206-13
- "Paycheck Analysis" program 187-92
- percentages, computation of 263, 268-69, 276-77
- "Pie Plot" program 268-75
- present value, of a dollar 349, 354
 - illustration of 349
- principal diagonal 235. *See also* matrices, inversion
- "Random Number Generator" program 201-5
- R-bar squared 311
- regression analysis 299-303, 308-12
- regression sum of squares 311
- relationships
 - exponential 322
 - linear 322
 - power 322
 - reciprocal 322
- "Relative and Cumulative Frequencies" program 262-67
- relative frequency 262-64, 276-77
- residual sum of squares 311
- residual variance 311-12
- rho 312
- row vector 219. *See also* matrices, dimensions
- R-squared 301, 363
- savings, financial 103-5, 120-22
- simple correlation coefficient 290-93
- "Simple Correlation Coefficients" program 290-98
- simple linear regression 299-303, 312, 325
- "Simple Linear Regression Analysis" program 299-307
- slope 300-302
- "Solitaire Checkers" program 51-60
- "Sort" program 197-200
- "Spelling Quiz" program 71-79
- standard deviation 258
- standard error of the estimate 301-2, 311-12
- standard errors 301, 311
- statistics 255-332
 - accuracy of fit statistics 301-2
 - analysis of variance 284-85
 - coefficient 290-301
 - constant 300-301
 - correlation coefficient 290-93
 - cumulative frequency 262-64
 - degrees of freedom 285
 - dependent variable 302, 310, 363
 - Durbin-Watson statistic 312
 - explanatory variable 302, 309, 363
 - frequency 263-64, 276-77
 - F-statistic 285, 302, 311
 - inverse relationship of variables 299
 - linear association 290, 292, 302, 312, 322
 - mean 202, 257-258
 - multiple linear regression 308-12
 - percentages 263, 268-69, 276-77
 - R-bar squared 311
 - regression analysis 299-303, 308-12
 - regression sum of squares 311
 - relative frequency 262-64, 276, 277
 - residual sum of squares 311
 - residual variance 311-12
 - R-squared 301, 363
 - simple correlation coefficient 290-93
 - simple linear regression 299-303, 312, 325
 - slope 300-302
 - standard deviation 258
 - standard error of the estimate 301-2, 311-12
 - standard errors 301, 311
 - total sum of squares 311
 - t-ratios 311
 - variables 290, 292-93, 299-303, 308-12, 322, 372
 - variance 202, 203, 258, 284-86
- taxes 103, 133-35, 187-88
- time-series forecasting 372-74
- "Time-Series Forecasting" program 372-81
- total sum of squares 311
- trapezoids 208, 210
- t-ratios 311
- "Utility Bill Audit" program 178-86
- variables 290, 292-93, 299-303, 308-12, 322, 372
- variance 202, 203, 258, 284-86
 - illustration of 202
- "Word Hunt" program 39-50
- "Weather Forecaster" program 159-69

If you've enjoyed the articles in this book, you'll find the same style and quality in every monthly issue of **COM-PUTE!** Magazine. Use this form to order your subscription to **COMPUTE!**.

For Fastest Service
Call Our **Toll-Free** US Order Line
800-334-0868
In NC call **919-275-9809**

COMPUTE!

P.O. Box 5406
Greensboro, NC 27403

My computer is:

- ☐ Commodore 64 ☐ TI-99/4A ☐ Timex/Sinclair ☐ VIC-20 ☐ PET
☐ Radio Shack Color Computer ☐ Apple ☐ Atari ☐ Other _____
☐ Don't yet have one...

- ☐ \$24 One Year US Subscription
☐ \$45 Two Year US Subscription
☐ \$65 Three Year US Subscription

Subscription rates outside the US:

- ☐ \$30 Canada and Foreign Surface Mail
☐ \$65 Foreign Air Delivery

Name _____

Address _____

City _____

State _____

Zip _____

Country _____

Payment must be in US funds drawn on a US bank, international money order, or charge card.

- ☐ Payment Enclosed ☐ Visa
☐ MasterCard ☐ American Express

Acct. No. _____

Expires _____

/

753199

Your subscription will begin with the next available issue. Please allow 4-6 weeks for delivery of first issue. Subscription prices subject to change at any time.



Home and Office

Apple II series computers are equally valuable in the school, the living room, or the office. And though the IIc is the latest of the series to evolve, the BASIC programs written for the II work on the IIc.

Easy BASIC Programs for the Apple contains over 40 such programs, all ready to type in and run—on any Apple II series computer. It's a wide-ranging collection, including thought-provoking games, home applications, mathematics, statistics, and business tools. Here's a sample of what you'll find:

- "Database Manager," a simple-to-use data storage and retrieval system.
- "Weather Forecaster," which allows you to make reliable short-range weather predictions.
- "Mosaic Puzzle," a re-creation of the predecessor to Rubik's Cube.
- "MatMan," a powerful mathematical tool that lets you perform a variety of matrix manipulations.
- "Pie Plot," which draws pie charts based on your information.
- "Computer Cash Register," to turn your Apple into a simple electronic register.

Whether you're using your Apple computer at school, at home, or in the office, you'll find that *Easy BASIC Programs for the Apple* offers something for almost any need. Like all COMPUTE! books, it provides clear instructions and high-quality, tested programs.

ISBN 0-942386-88-4